AD-A277 084

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: U

1b. RESTRICTIVE MARKINGS: NA

2a. SECURITY CLASSIFICATION AUTHORITY: NA

3. DISTRIBUTION/AVAILABILITY OF REPORT: DISTRIBUTION UNLIMITED

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE: NA

4. PERFORMING ORGANIZATION REPORT NUMBER(S):

5. MONITORING ORGANIZATION REPORT NUMBER(S): NA

6a. NAME OF PERFORMING ORGANIZATION: Penn State University

6b. OFFICE SYMBOL (If applicable): NA

7a. NAME OF MONITORING ORGANIZATION: OFFICE OF NAVAL RESEARCH

6c. ADDRESS (City, State, and ZIP Code):
Decision & Control Systems Laboratory
Department of Electrical Engineering
University Park, PA 16802

7b. ADDRESS (City, State, and ZIP Code):
800 N. quincy st.
Arlington, VA 22217-5000

8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Office of Naval Research

8b. OFFICE SYMBOL (If applicable): ONR

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: N00014-86-k-0515

8c. ADDRESS (City, State, and ZIP Code):
800 N. Quincy St.
Arlington, VA 22217-5000

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
61153N	RR04106		

94-08498

11. TITLE (Include Security Classification): INTEGRATED DECISION, ESTIMATION AND COMMUNICATION THEORIES

12. PERSONAL AUTHOR(S): Dr. Stelios C. A. Thomopoulos

13a. TYPE OF REPORT: Final

13b. TIME COVERED: FROM 05/86 TO 05/90

14. DATE OF REPORT (Year, Month, Day): 08/26/93

15. PAGE COUNT: 10 + References

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
08		

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Distributed, Decision, Fusion, Estimation, Communication

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The objectives of this program were to investigate the synergies among the decision, estimation and communication aspects of a distributed multisensor system. Hence, the effort in this project was primarily concentrated in the development of a coherent framework that would allow the development of a coherent theory of distributed decision and incorporate estimation and communication aspects. In this context, a Neyman-Pearson theory for distributed decison fusion was developed. The effect of communcations and topological aspects in the structure and performance of the opitmal distributed decision fusion have been investigated. The opitmal distributed Neyman-Pearson decision fusion has been derived for the ideal case, and in cases where transmission delays, channel erors, and sensor misalignment are present. Other issues involved in the design of a distributed decision fusion system, such as intersensor correlation and multiresolution detection have also been investigated. A Generalized Evidence Processing theory that extends and to certain extend unifies, the Bayesian and Dempster-Shafer theories has been developed. A systematic framework for the data fusion analysis and synthesis has also been developed and tested with experimental data successfully.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION: (U)

22a. NAME OF RESPONSIBLE INDIVIDUAL: Dr. E. Schnell or M. Marron or M. Haggard

22b. TELEPHONE (Include Area Code): (202) 696-4760

22c. OFFICE SYMBOL: ONR

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

INTEGRATED DECISION, ESTIMATION AND COMMUNICATION THEORIES

FINAL REPORT FOR CONTRACT N00014-86-k-0515

prepared by

Stelios C. A. Thomopoulos

Decision and Control Systems Laboratory
Department of Electrical Engineering
The Pennsylvania State University
University Park, PA 16802

Tel. (814) 865-3744; Fax: (814) 865-7065
e-mail: sct@ecl.psu.edu

August 24, 1993

INTEGRATED DETECTION, ESTIMATION AND COMMUNICATION THEORIES

The objectives of this program were to investigate the synergies among the decision, estimation and communication aspects of a distributed multisensor system. The effort in this project was hence concentrated primarily on the development of a coherent framework for data fusion and the development of a coherent theory of distributed decision capable of incorporating estimation and communication aspects.

A fair amount of effort was focused on the development of a distributed decision fusion theory. In this context a Neyman-Pearson type theory was developed for the distributed decison fusion problem. The theory has been developed for the binary hypothesis testing problem with both binary and M-ary quantized decisions at the local (sensor) level. The thoery has established that, under statistical independence, the optimal fusion configuration consists of binary (or M-ary) level likelihood quantizers at the sensor level, and a binary Neyman-Pearson test at the fusion. Variants of this optimal Neyman-Pearson solution have been investigated and the optimal (in the Bayesian or N-P sense) solutions were obtained in the presence of propagation delays in the transmission of the decisions from the sensor to fusion, presence of error in the fused data, and in the presence of sensor misalignment and communication constraints in the provision of information.

Other issues involved in the design of a distributed decision fusion system, such as intersensor correlation and multiresolution detection have been investigated.

A large number of publications have been emerged from this project and have appeared in scattered journals or conference proceedings. A sample of a few publiactions is attached.

The success of this program has led to the teaming of the P.I. with Calspan and Crumman Cooperations and the submission of a proposal for Pre Detection Fusion to Rome ADC. The success of the Pre Detection Fusion program. The contract was awarded to our team. The project has ended successfully. The acquired experience from this project, the first contolled environment data fusion project, has been invaluable.

The basis of distributed decision theory has been expanded to more genral fusion concepts. As a result, a Generalized Evidence Processing (GEP) theory was developed. The developed theory attempts to reconciliate the Bayesian with the Dempster-Shafer theory. Numerical comparisons between the GEP and conventional distributed fusion algorithms, highlight the superior performance of GEP as compared to the conventional distibuted decision theory.

A list of publications that resulted from this project, a list of recent publications that relate to this project directly, and a sample of the main publications that emerged from the project follow.

List of publications from this project

A. Refereed Journals and Proceedings

S. C. A. Thomopoulos, R. Viswanathan and D. K. Bougoulias, "Optimal Decision Fusion in Multiple Sensor Systems," IEEE Transactions on Aerospace and Electronic Systems, Vol. 23, No. 5, Sept. 1987, pp. 644-653.

R. Viswanathan, S. C. A. Thomopoulos, and R. Tumuluri, "Optimal Serial Distributed Decision Fusion," IEEE Transactions on Aerospace and Electronic Systems, Vol. 24, No. 4, July 1988, pp. 366-376.

S. C. A. Thomopoulos, R. Viswanathan, and D. K. Bougoulias, "Optimal Distributed Decision Fusion," IEEE Transactions on Aerospace and Electronic Systems, Vol. 25, No. 5, Sept. 1989, pp. 761-764.

S. C. A. Thomopoulos, "Sensor Integration and Data Fusion," Invited paper in special issue on Sensor Integration and Data Fusion for Robotic Systems, Journal of Robotic Systems, Vol. 7, No. 3, 1990, pp. 337-372.

S. C. A. Thomopoulos and N. Okello, "Distributed Detection with Consulting Sensors and Communication Cost," IEEE Transactions on Automatic Control, Vol. 37, no. 9, September 1992, pp. 1398-1405.

S. C. A. Thomopoulos and L. Zhang, "Distributed Decision Fusion with Networking Delays and Channel Errors," Information Sciences: An International Journal, Vol. 66, nos. 1 & 2, December 1, 1992, pp. 91-118.

S. C. A. Thomopoulos and N. N. Okello, "Distributed and Centralized Multi-Sensor Detection with Misaligned Sensors," Information Sciences: An International Journal, to appear.

I. N. M. Papadakis and S. C. A. Thomopoulos, "Hypothesis Testing using Structured Networks," IEEE Transactions on Automatic Control, to appear.

S. C. A. Thomopoulos, D. K. Bougoulias and C.-D. Wann, "**Dignet:** An Unsupervised Clustering Algorithm for Centralized and Distributed Pattern Recognition, Classification, and Hypothesis Testing," IEEE Transactions on Aerospace and Electronic Systems, to appear.

S. C. A. Thomopoulos, R. Viswanathan, and D. P. Bougoulias, "Optimal Decision Fusion in Multiple Sensor Systems," 24th Allerton Conference, Allerton House, Monticello, IL, Oct. 1-3, 1986.

R. Viswanathan, S. C. A. Thomopoulos, and V. Aalo, "Distributed Detection with Correlated Sensor Noise, 25th Allerton Conference, Allerton House, Monticelli, IL, Sept. 30-Oct. 2, 1987.

V. Aalo, R. Viswanathan, and S. C. A. Thomopoulos, "A Study of Distributed Detection with Correlated Sensor Noise," Proceedings of IEEE GLOBECOM '87, Tokyo, Japan, Nov. 15-18, 1987.

S. C. A. Thomopoulos, "Optimal and Suboptimal Decision Fusion, 35th SIAM Meeting, Society for Industrial and Applied Mathematics, Denver, Colorado, Oct. 12-15, 1987.

S. C. A. Thomopoulos, L. Zhang, and R. Viswanathan, "Distributed Detection and Networking," Symposium on Innovative Science and Technology, SPIE '88, Los Angeles, CA, January 10-15, 1987.

R. Viswanathan, S. C. A. Thomopoulos, and R. Tumuluri, "Sequential Decision in Multiple Sensor Fusion," Proceedings of 21st CISS, The Johns Hopkins University, March 25-27, 1987.

S. C. A. Thomopoulos, and N. Okello, "Distributed Detection with Consulting Sensors and Communication Cost," SPIE's 1988 Technical Symposium on Optics, Electro-Optics and Sensors, Orlando, FL, April 4-8, 1988.

S. C. A. Thomopoulos, and L. Zhang, "Distributed Decision Fusion with Networking Delays and Channel Errors," SPIE Proceedings, Vol. 931, Sensor Fusion, (1988), pp. 154-160.

S. C. A. Thomopoulos and L. Zhang, "Distributed Filtering with Random Sampling and Delay," SPIE Proceedings, Vol. 931, Sensor Fusion, (1988), pp. 31-40.

S. C. A. Thomopoulos, D. K. Bougoulias, and L. Zhang, "Optimal and Suboptimal Distributed Decision Fusion, SPIE Proceedings, Vol. 931, Sensor Fusion, (1988), pp. 26-30.

S. C. A. Thomopoulos, R. Viswanathan and D. K. Bougoulias, "Optimal and Suboptimal Distributed Decision Fusion," 22nd Annual Conference on Information Sciences and Systems, Princeton University, March 16-18, 1988.

S. C. A. Thomopoulos and N. Okello, "Distributed Detection with Consulting Sensors and Communication Cost," 22nd Annual Conference on Information Sciences and Systems, Princeton University, March 16-18, 1988.

S. C. A. Thomopoulos and L. Zhang, "Networking in Distributed Decision Fusion," American Control Conference ACC '88, Atlanta, GA, June 15-17, 1988.

S. C. A. Thomopoulos and N. Okello, "Decision Fusion with Consulting Sensors," American Control Conference ACC '88, Atlanta, GA, June 15-17, 1988.

S. C. A. Thomopoulos and N. Okello, "Distributed Detection with Mismatched Sensors," SPIE 1988 Cambridge Symposium on Advances in Intelligent Robotics Systems, Boston MA, October 1988.

S. C. A. Thomopoulos and L. Zhang, "Distributed Filtering with Random Sampling and Delay," Conference on Decision and Control, CDC '88, Austin, Texas, December 7-10, 1988.

S. C. A. Thomopoulos, "Sensor Integration and Data Fusion," SPIE Proceedings on Advances in Intelligent Robotics Systems, Sensor Fusion II: Human and Machine Strategies, Vol. 1198, pp. 178-191.

S. C. A. Thomopoulos and L. Nilsson, "Object Tracking for Sequences of Images Using Sterea Camera," SPIE Proceedings on Advances in Intelligent Robotics Systems, Sensor Fusion II: Human and Machine Strategies, Vol. 1198, pp. 156-169.

S. C. A. Thomopoulos, "Theories in Distributed Decision Fusion: Comparison and Generalization," SPIE Proceedings, Vol. 1383, Sensor Fusion III, Nov. 1990.

S. C. A. Thomopoulos and D. K. Bougoulias, "DIGNET: A Self-Organizing Neural Network for Automatic Pattern Recognition and Classification," Proceeding of CISS, Johns Hopkins, March 1991.

S. C. A. Thomopoulos and D. K. Bougoulias, "DIGNET: A Self-Organizing Neural Network for Automatic Pattern Recognition and Classification," SPIE's OE/Aerospace Sensing, Orlando, FL, 1-5 April, 1991.

B. Books or parts of books

S. C. A. Thomopoulos, "Decision and Evidence Fusion in Sensor Integration," Advances in Control and Dynamic Systems, Ed. C. T. Leondes, Vol. 49, Part 5 of 5, pp. 339-412, Academic Press, Nov. 1991.

S. C. A. Thomopoulos, R. Viswanathan and D. K. Bougoulias, "Optimal Decision Fusion in Multiple Sensor Systems," in Multisensor Integration and Fusion for Intelligent Machines and Systems, Editors R. C. Luo and M. G. Kay, Ablex Publ., Computer Engineering and Computer Science Series, in press, scheduled for February 1993.

S. C. A. Thomopoulos, R. Viswanathan and D. K. Bougoulias, "Optimal Distributed Decision Fusion," in Multisensor Integration and Fusion for Intelligent Machines and Systems, Editors R. C. Luo and M. G. Kay, Ablex Publ., Computer Engineering and Computer Science Series, in press, scheduled for February 1993.

R. Viswanathan, S. C. A. Thomopoulos, and R. Tumuluri, "Optimal Serial Distributed Decision Fusion," in Multisensor Integration and Fusion for Intelligent Machines and Systems, Editors R. C. Luo and M. G. Kay, Ablex Publ., Computer Engineering and Computer Science Series, in press, scheduled for February 1993.

RECENT PUBLICATIONS THAT RELATE TO THE *INTEGRATED DETECTION, ESTIMATION AND COMMUNIATION THEORIES*

S. C. A. Thomopoulos, "Asymptotic Performance of Distributed Decision Fusion System with Binary and Multilevel Memoryless Nonlinearities," Sensor Fusion IV: Control Paradigms and Data Structures, SPIE Proceedings, Boston, Nov. 1992, pp. 443-451.

S. C. A. Thomopoulos and D. K. Bougoulias, "DIGNET: A Self-Organizing Neural Network for Automatic Pattern Recognition, Classification and Data Fusion," Sensor Fusion IV: Control Paradigms and Data Structures, SPIE Proceedings, Boston, Nov. 1992, pp. 478-495.

S. C. A. Thomopoulos, I. N. M. Papadakis, H. Sahinolglou, and N. N. Okello, "Centralized and Distributed Hypothesis Testing with Structured Adaptive Networks and Perception-like Neural Networks," Sensor Fusion IV: Control Paradigms and Data Structures, SPIE Proceedings, Boston, Nov. 1992, pp. 35-51.

S. C. A. Thomopoulos and D. K. Bougoulias, "DIGNET: A Self-Organizing Neural Network for Automatic Pattern Recognition and Classification," 30th IEEE Conference on Decision and Control (CDC '91), Brighton, U.K., Dec. 11-13, 1991, pp. 853-858.

S. C. A. Thomopoulos and D. K. Bougoulias, "DIGNET: A Self-Organizing Neural Network for Automatic Pattern Recognition and Classification," International Joint Conference on Neural Networks (IJCNN) '91, Singapore, Nov. 18-21, 1991, pp. 2683-2692.

T. W. Hilands and S. C. A. Thomopoulos, "Estimating the Parameters of Exponentially Damped Sinusoids using Nonlinear Filters," Proceedings of the 1992 CISS, Princeton University, March 1992.

I. N. M. Papadakis and S. C. A. Thomopoulos, "Hypothesis Testing using Structured Networks," American Control Conference (ACC) '92, Chicago, Il, June 24-26, 1992.

T. W. Hilands and S. C. A. Thomopoulos, "High-Order Moments Nonlinear Filtering Methods with applications to Harmonic Retrieval," Canadian Conference on Electrical and Computer Engineering, CCECE-92, Toronto, Canada, Sept. 13-16, 1992.

S. C. A. Thomopoulos and T. W. Hilands, "Nonlinear Techniques for Model Order Selection in Gaussian and non-Gaussian Noise," Canadian Conference on Electrical and Computer Engineering, CCECE-92, Toronto, Canada, Sept. 13-16, 1992.

S. C. A. Thomopoulos and T. W. Hilands, "Joint Detection/Estimation for Model Order Selection in Gaussian and non-Gaussian Noise," Milcom '92, San Diego, CA, Oct. 11-14, pp. 613-617.

T. W. Hilands and S. C. A. Thomopoulos, "Nonlinear Filtering Methods for Harmonic

Retrieval," Mikcom '92, San Diego, CA, Oct. 11-14, 1992, pp. 1081-1085.

S. C. A. Thomopoulos and T. W. Hilands, "Nonlinear Adaptive Detection/Estimation for Single and Multiple Radar Processing," Mikcom '92, San Diego, CA, Oct. 11-14, 1992, pp. 1076-1080.

M. D. Reger and S. C. A. Thomopoulos, "High Resolution Target Detection in the Presence of Multipath," '93 IEEE Regional Conference on Aerospace Control Systems, Westlake Village, CA, May 25-27, 1993.

S. C. A. Thomopoulos and T. W. Hilands, "High Resolution Model Order Selection and Parameter Estimation in Gaussian and Non-Gaussian Noise Using JDEF," '93 IEEE Regional Conference on Aerospace Control Systems, Westlake Village, CA, May 25-27, 1993.

L. Weiss and S. C. A. Thomopoulos, "Target Detection and Localization from Bearing-Only and Bearing/Range Measurements," '93 IEEE Regional Conference on Aerospace Control Systems, Westlake Village, CA, May 25-27, 1993.

S. C. A. Thomopoulos and N. N. Okello, "Design of a Robust Multi-Radar Distributed Data Fusion System," '93 IEEE Regional Conference on Aerospace Control Systems, Westlake Village, CA, May 25-27, 1993.

T. W. Hilands and S. C. A. Thomopoulos, "High Resolution Multi-Radar Data Fusion Using JDEF," '93 IEEE Regional Conference on Aerospace Control Systems, Westlake Village, CA, May 25-27, 1993.

C.-D. Wann and S. C. A. Thomopoulos, "Comparative Study of Self-Organizing Neural Networks in Signal Detection and Classification," '93 IEEE Regional Conference on Aerospace Control Systems, Westlake Village, CA, May 25-27, 1993.

B. H. Chen and S. C. A. Thomopoulos, "High Resolution Image Segmentation Using JDEF," '93 IEEE Regional Conference on Aerospace Control Systems, Westlake Village, CA, May 25-27, 1993.

S. C. A. Thomopoulos, I. Kadar, L. Lovas, and N. N. Okello, "Design of Multisensor data fusion system for target detection," SPIE Proceedings, Vol. 1955, Orlando, FL, April 12-14, 1993.

T. W. Hilands and S. C. A. Thomopoulos, "Model-based JointDetection/Estimation Approach for Multisensor Data Fusion," SPIEProceedings, Vol. 1955, Orlando, FL, April 12-14, 1993.

B. H. Chen and S. C. A. Thomopoulos, "Image Restoration by Non-Homogeneous G-M Field Modeling and Adaptive Kalman Filtering," '93 IEEE International Symp. on Circuits and Systems, Chicago, IL, May 3-6,1993.

SAMPLE OF MAIN PUBLICATIONS THAT RESULTED FROM THE PROJECT

INTEGRATED DETECTION, ESTIMATION AND COMMUNICATION THEORIES

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION U	1b. RESTRICTIVE MARKINGS NA
2a. SECURITY CLASSIFICATION AUTHORITY NA	3. DISTRIBUTION/AVAILABILITY OF REPORT DISTRIBUTION UNLIMITED
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE NA	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) NA
6a. NAME OF PERFORMING ORGANIZATION Penn State University 6b. OFFICE SYMBOL (If applicable) NA	7a. NAME OF MONITORING ORGANIZATION OFFICE OF NAVAL RESEARCH
6c. ADDRESS (City, State, and ZIP Code) Decision & Control Systems Laboratory Department of Electrical Engineering University Park, PA 16802	7b. ADDRESS (City, State, and ZIP Code) 800 N. quincy st. Arlington, VA 22217-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Research 8b. OFFICE SYMBOL (If applicable) ONR	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-86-k-0515
8c. ADDRESS (City, State, and ZIP Code) 800 N. Quincy St. Arlington, VA 22217-5000	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO: 61153N PROJECT NO.: RR04106 TASK NO: WORK UNIT ACCESSION NO:

11. TITLE (Include Security Classification)
INTEGRATED DECISION, ESTIMATION AND COMMUNICATION THEORIES

12. PERSONAL AUTHOR(S)
Dr. Stelios C. A. Thomopoulos

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 05/86 TO 05/90	08/26/93	10 + References

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Distributed, Decision, Fusion, Estimation, Communication
08			

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The objectives of this program were to investigate the synergies among the decision, estimation and communication aspects of a distributed multisensor system. Hence, the effort in this project was primarily concentrated in the development of a coherent framework that would allow the development of a coherent theory of distributed decision and incorporate estimation and communication aspects. In this context, a Neyman-Pearson theory for distributed decison fusion was developed. The effect of communcations and topological aspects in the structure and performance of the opitmal distributed decision fusion have been investigated. The opitmal distributed Neyman-Pearson decision fusion has been derived for the ideal case, and in cases where transmission delays, channel erors, and sensor misalignment are present. Other issues involved in the design of a distributed decision fusion system, such as intersensor correlation and multiresolution detection have also been investigated. A Generalized Evidence Processing theory that extends and to certain extend unifies, the Bayesian and Dempster-Shafer theories has been developed. A systematic framework for the data fusion analysis and synthesis has also been developed and tested with experimental data successfully.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION (U)
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. E. Schnell or M. Marron or M. Haggard	22b. TELEPHONE (Include Area Code) (202) 696-4760 22c. OFFICE SYMBOL ONR

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Optimal Decision Fusion in Multiple Sensor Systems

STELIOS C.A. THOMOPOULOS, Member, IEEE
RAMANARAYANAN VISWANATHAN, Member, IEEE
DIMITRIOS C. BOUGOULIAS, Student Member, IEEE
Southern Illinois University

The problem of optimal data fusion in the sense of the Neyman-Pearson (N-P) test in a centralized fusion center is considered. The fusion center receives data from various distributed sensors. Each sensor implements a N-P test individually and independently of the other sensors. Due to limitations in channel capacity, the sensors transmit their decision instead of raw data. In addition to their decisions, the sensors may transmit one or more bits of quality information. The optimal, in the N-P sense, decision scheme at the fusion center is derived and it is seen that an improvement in the performance of the system beyond that of the most reliable sensor is feasible, even without quality information, for a system of three or more sensors. If quality information bits are also available at the fusion center, the performance of the distributed decision scheme is comparable to that of the centralized N-P test. Several examples are provided and an algorithm for adjusting the threshold level at the fusion center is provided.

Manuscript received July 17, 1986; revised March 3, 1987.

This research is sponsored by the SDIO/IST and managed by the Office of Naval Research under Grant N00014-86-K-0515.

Authors' address: Dept. of Electrical Engineering, College of Engineering and Technology, Southern Illinois University, Carbondale, IL 62901.

0018-9251/87/0900-0644 $1.00 © 1987 IEEE

I. INTRODUCTION

The problem of data fusion in a central decision center has attracted the attention of several investigators due to the increasing interest in the deployment of multiple sensors for communication and surveillance purposes. Because of a limited transmission capacity, the sensors are required to transmit their decision (with or without quality information bits) instead of the raw data the decisions are based upon. A centralized fusion center is responsible for combining the received information from the various sensors into a final decision.

Tenney and Sandell [1] have treated the Bayesian detection problem with distributed sensors. However, they did not consider the design of data fusion algorithms. Sadjadi [2] has considered the problem of general hypothesis testing in a distributed environment and has provided a solution in terms of a number of coupled equations. The decentralized sequential detection problem has been investigated in [3–5]. Chair and Varshney [6] have considered the problem of data fusion in a central center when the data that the fusion center receives consist of the decisions made by each sensor individually and independently from each other. They derive the optimal fusion rule for the likelihood ratio (LR) test. It turns out that the sufficient statistics for the LR test is a weighted average of the decisions of the various sensors with weights that are functions of the individual probabilities of false alarm P_F and the probabilities of detection P_D. However, the maximum a-posteriori (MAP) test or the LR test require either exact knowledge of the a-priori probabilities of the tested hypotheses or the assumption that all hypotheses are equally likely. However, if the Neyman-Pearson (NP) test is employed at each sensor, the same test must be used to fuse the data at the fusion center, in order to maximize the probability of detection for fixed probability of false alarm.

We derive the optimal decision scheme when the N-P test is used at the fusion center. The optimal decision scheme, in the N-P sense, is derived: 1) for cases where the various sensors transmit exclusively their decisions to the fusion center, and 2) for cases where the various sensors transmit quality bits along with their decisions indicating the degree of their confidence in their decision.

II. DECISION FUSION WITH THE NEYMAN-PEARSON TEST

Consider the problem of two hypotheses testing with H_1 designating one hypothesis and H_0 the alternative. Assume that the prior probabilities on the two hypotheses are not known. A number of sensors N receive observations and independently implement the N-P test. Let u_j designate the decision of the jth sensor having taken into account all the observations available to this sensor at the time of the decision. If the decision of the jth sensor favors hypothesis H_1, the sensor sets u_j =

+1, otherwise it sets $u_j = -1$. Every sensor transmits its decision to the fusion center, so that the fusion center has all N decisions available for processing at the time of the decision making. Let (P_{F_j}, P_{D_j}) designate the pair of the probability of false alarm and the probability of detection at which the jth sensor operates and implements the N-P test. The fusion center implements the N-P test using all the decisions that the individual sensors have communicated, i.e., it formulates the LR test:

$$\Lambda(u) = \frac{P(u_1, u_2, \ldots, u_N|\mathrm{H}_1)}{P(u_1, u_2, \ldots, u_N|\mathrm{H}_0)} \underset{\mathrm{H}_0}{\overset{\mathrm{H}_1}{\gtrless}} t \qquad (1)$$

where $u = (u_1, u_2, \ldots, u_N)$ is a $1 \times N$ row vector with entries the decisions of the individual sensors, and t the threshold to be determined by the desirable probability of false alarm at the fusion center P_F^f, i.e.,

$$\sum_{\Lambda(u)>t} P(\Lambda(u)|\mathrm{H}_0) = P_F^f \qquad (2)$$

Since the decisions of each sensor are independent from each other, the LR test (1) gives

$$\Lambda(u) = \prod_{i=1}^{N} \frac{P(u_i|\mathrm{H}_1)}{P(u_i|\mathrm{H}_0)} \underset{\mathrm{H}_0}{\overset{\mathrm{H}_1}{\gtrless}} t \qquad (3)$$

from which the result in [6] is readily obtained. In order to implement the N-P test we need to compute $P(\Lambda(u)|\mathrm{H}_0)$. However, due to the independence assumption, it is easier to obtain the distribution $P(\log \Lambda(u)|\mathrm{H}_0)$ which can be expressed as the convolution of the individual $P(\log \Lambda(u_i)|\mathrm{H}_0)$. Thus, it follows from (3):

$$P(\log \Lambda(u)|\mathrm{H}_0) = P(\log \Lambda(u_1)|\mathrm{H}_0) * \cdots * P(\log \Lambda(u_N)|\mathrm{H}_0). \qquad (4)$$

The LR $\Lambda(u_i)$ assumes two values. Either $(1 - P_{D_i})/(1 - P_{F_i})$ when $u_i = 0$ with probability $1 - P_{F_i}$ under hypothesis H_0 and probability $1 - P_{D_i}$ under hypothesis H_1, or, P_{D_i}/P_{F_i} when $u_i = 1$ with, probability P_{F_i} under hypothesis H_0 and probability P_{D_i} under hypothesis H_1. Hence, we can write

$$P(\log \Lambda(u_i)|\mathrm{H}_0) = (1 - P_{F_i})\,\delta\left(\log \Lambda(u_i) - \log \frac{1 - P_{D_i}}{1 - P_{F_i}}\right) + P_{F_i}\,\delta\left(\log \Lambda(u_i) - \log \frac{P_{D_i}}{P_{F_i}}\right) \qquad (5)$$

and

$$P(\log \Lambda(u_i)|\mathrm{H}_1) = (1 - P_{D_i})\,\delta\left(\log \Lambda(u_i) - \log \frac{1 - P_{D_i}}{1 - P_{F_i}}\right) + P_{D_i}\,\delta\left(\log \Lambda(u_i) - \log \frac{P_{D_i}}{P_{F_i}}\right) \qquad (6)$$

where

$$\delta(x) = \begin{cases} 1 & \text{for } x = 0 \\ 0 & \text{for } x \neq 0. \end{cases}$$

At the fusion center, the probability of false alarm

$$P_F^f = \sum_{\Lambda(u)>t^*} P(\Lambda(u)|\mathrm{H}_0) \qquad (7)$$

where t^* is a threshold chosen to satisfy (7) for a given P_F^f. Similarly, the probability of detection at the fusion center

$$P_D^f = \sum_{\Lambda(u)>t^*} P(\Lambda(u)|\mathrm{H}_1). \qquad (8)$$

A. Similar Sensors

When all the sensors are similar and operate at the same level of probability of false alarm and probability of detection, i.e., $P_{F_i} = P_{F_j} = P_F$ and $P_{D_i} = P_{D_j} = P_D$ for every i and j, all the probability distributions in (3) are the same and the N-P test leads to the following scheme at the fusion center. (Expression similar to (9) and (10) were obtained in [6] for the LR test.)

$$\sum_{i=1}^{N} a_i u_i \underset{\mathrm{H}_0}{\overset{\mathrm{H}_1}{\gtrless}} t \qquad (9)$$

where

$$a_i = \begin{cases} \log\left(\dfrac{P_D}{P_F}\right), & \text{if } u_i = +1, i = 1, \ldots, N \\[2ex] \log\left(\dfrac{1 - P_F}{1 - P_D}\right), & \text{if } u_i = -1, i = 1, \ldots, N. \end{cases} \qquad (10)$$

If k out of the N decisions favor hypothesis H_1, (9) can be rewritten as

$$k\left(\log\left[\frac{P_D(1 - P_F)}{P_F(1 - P_D)}\right]\right) \underset{\mathrm{H}_0}{\overset{\mathrm{H}_1}{\gtrless}} t + N \log\left(\frac{1 - P_F}{1 - P_D}\right) \qquad (11)$$

For all sensible tests, though, $P_F < P_D$. Hence, $\log \dfrac{P_D(1 - P_F)}{P_F(1 - P_D)} > 0$ and the N-P test becomes

$$k \underset{\mathrm{H}_0}{\overset{\mathrm{H}_1}{\gtrless}} t^* \qquad (12)$$

where t^* is some threshold to be determined so that a certain overall false alarm probability P_F^f is attained at the fusion center.

The random variable k has a binomial distribution with parameters N and $\mathrm{P_F}$ under H_0 and parameters N and $\mathrm{P_D}$ under H_1. Hence, P_F^f and the overall probability of detection P_D^f are given by

$$P'_F = \sum_{i=\lceil t^* \rceil}^{N} \binom{N}{i} P_F^i (1 - P_F)^{N-i} \tag{13}$$

$$P'_D = \sum_{i=\lceil t^* \rceil}^{N} \binom{N}{i} P_D^i (1 - P_D)^{N-i} \tag{14}$$

where $\lceil t^* \rceil$ indicates the smallest integer exceeding t^*. The threshold t^* must be determined so that (12) gives an acceptable overall probability of false alarm.

For the configuration of N sensors, we are interested to know whether the N-P test can provide a (P'_F, P'_D) pair such that

$$P'_F \leqq \min_{i \in N}\{P_F\} \quad \text{and} \quad P'_D > \max_{i \in N}\{P_D\} \tag{15}$$

where (P_D, P_F) is the N-P test level for sensor i, $i = 1, \ldots, N$.

The next Theorem shows that condition (15) can be satisfied if the randomized N-P test is used at the fusion center, the number of sensors N is greater than two, and all the sensors are characterized by the same (P_F, P_D) pair.

Theorem. In a configuration of N similar sensors, all operating at the same $(P_F, P_D) = (p, q)$, the randomized N-P test at the fusion center can provide a (P'_F, P'_D) satisfying (15) if $N \geqq 3$.

More precisely, for $N \geqq 3$, the randomized N-P test can be fixed so that

$$P'_F = P_F = p \quad \text{and} \quad P'_D > P_D = q \tag{16}$$

where P_F and P_D are the probability of false alarm and probability of detection at the individual sensors.

Proof. First we show that for $N = 2$, condition (15) cannot be satisfied with the second inequality as a strict one. Then we prove that for $N = 3$, the randomized N-P test satisfies condition (15). By using the fact that for fixed probability of false alarm, the probability of detection at the fusion center is maximized by the N-P test among all mappings from the observation space into the decision space, we prove by induction that condition (15) is satisfied for all $N \geqq 3$.

Let $N = 2$ and $(P_F, P_D) = (p, q)$ for both sensors. Using (4), (5), (6), (9), and (10), the LR distributions at the fusion center under hypothesis H_0 and H_1 are plotted for the reader's convenience in Figs. 1 and 2, respectively. Since for all p in (0, 1)

$$p^2 < p < 2p(1-p) + p^2 \tag{17}$$

it follows that, in order to satisfy $P'_F = p$, the randomized N-P test must be used at the fusion center with threshold $q(1-q)/p(1-p)$ and randomizing factor ω defined by

$$p^2 + \omega 2p(1-p) = p \tag{18}$$

where $0 < \omega < 1$. Solving (18) we obtain $\omega = 0.5$, independent of p. Since P'_D is determined by an expression symmetric to (18) (see Figs. 1 and 2), $P'_D = q$ for $\omega = 0.5$. Hence, neither condition (16) nor condition (15) (which is more restrictive) can be satisfied for $N = 2$.

Let $N = 3$. The distributions of the LR under H_0 and H_1 are given in Figs. 3 and 4, respectively. From Fig. 3,

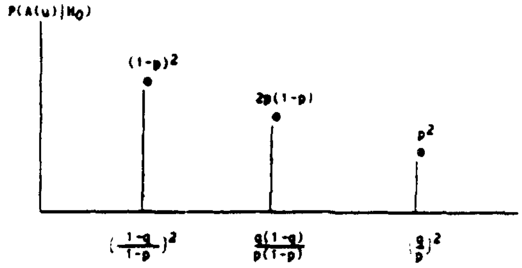


Fig. 1. Distribution of LR at fusion center under hypothesis H_0 for two similar sensor system, $N = 2$.

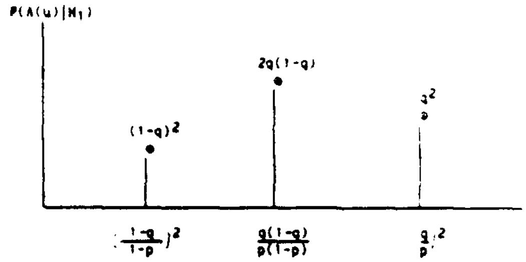


Fig. 2. Distribution of LR at fusion center under hypothesis H_1 for two similar sensor system, $N = 2$.

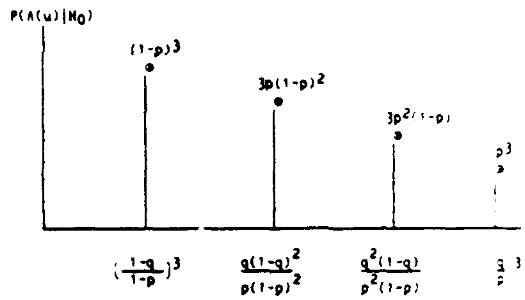


Fig. 3. Distribution of LR at fusion center under hypothesis H_0 for three similar sensor system, $N = 3$.

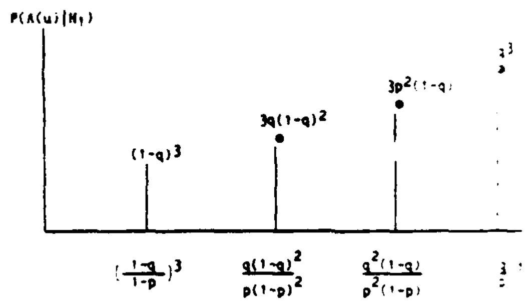


Fig. 4. Distribution of LR at fusion center under hypothesis H_1 for three similar sensor system, $N = 3$.

if the threshold at the fusion center is set at $q^2(1-q)/p^2(1-p)$.

$$P_F^f = p^3 + 3p^2(1-p) < p \tag{19}$$

for $0 < p < 0.5$. The left-hand side (LHS) of inequality (19) is greater than p for $p > 0.5$. Hence, since $P_F < 0.5$, the randomized N-P test that satisfies (15) at the fusion center is determined by

$$p^3 + 3p^2(1-p) + \omega 3p(1-p)^2 = p \tag{20}$$

from which

$$\omega = \frac{1}{3} - \frac{p}{3(1-p)}. \tag{21}$$

Hence, ω is a positive fraction for $0 < p < 0.5$.

Since P_D^f at the fusion center is given by an expression similar to (20) (see Fig. 4), with q in place of p, and $q > 0.5$, it follows from (20) that $P_D^f > q$, which proves the Theorem for $N = 3$.

Assume that the randomized N-P test satisfies condition (16) for an arbitrary number of sensors N. We show that it also satisfies the condition for $N + 1$, and thus complete the induction and the proof of the Theorem.

Let $U_N = \{u_1, u_2, \ldots, u_N\}$ designate the set of decisions from the N sensors that are available at the fusion center. All the sensors operate at the same level (p, q). Let $f_N(U_N)$ designate some decision rule at the fusion center operating at fixed probability of false alarm p. Let $f_N^{\text{N-P}}(U_N)$ designate the randomized N-P test at the fusion center at level p. For fixed probability of false alarm, the probability of detection at the fusion center (power of test) is maximized for the *N-P decision rule* among all possible decision rules.

Let $U_{N+1} = \{U_N, u_{N+1}\}$ designate the decision ensemble of $N + 1$ similar sensors all operating at the same level (p, q). Then by choosing $f_{N+1}(U_{N+1}) = f_N^{\text{N-P}}(U_N)$,

$$\begin{aligned} P_D(f_{N+1}^{\text{N-P}}(U_{N+1})) &= \max_{f_{N+1}(U_{N+1})} P_D(f_{N+1}(U_{N+1})) \\ &\geq P_D(f_N^{\text{N-P}}(U_N)) \end{aligned} \tag{22}$$

from which it follows that

$$P_{D,N+1}^f \geq P_{D,N}^f > q. \tag{23}$$

Thus the induction is complete and so is the proof of the Theorem.

Consider a system of four sensors $N = 4$ all operating at $P_F = 0.05$ and $P_D = 0.95$. If $t_f^* = 2$, from the binomial cumulative table we get $P_F^f = 0.014$ and $P_D^f = 0.9995$ at the fusion center, i.e., a considerable improvement in the performance of the overall system. From the binomial cumulative table it can be seen that at least three sensors are required for the decision fusion scheme to improve the performance of the system, as the Theorem suggests.

To assess the performance of the fusion scheme further, we compare it with the best centralized scheme, the N-P test which utilizes raw data, not decisions, from the different sensors. The loss associated with the use of decisions instead of raw data at the fusion center, is assessed by means of a simple example. Let a single observation from each of the four ($N = 4$) sensors be distributed normally (see Fig. 5) as

$$\begin{aligned} r_i &\sim G(0, 1), \quad \text{under } H_0 \\ &\sim G(S, 1), \quad \text{under } H_1. \end{aligned}$$

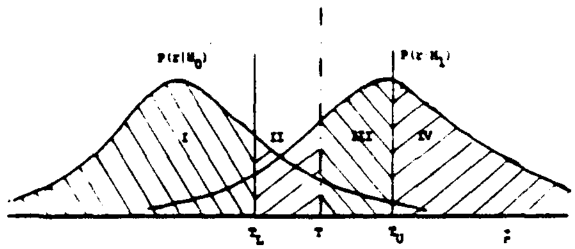


Fig. 5 Data distribution at each sensor under hypotheses H_0 and H_1, and confidence regions. Threshold is indicated by T. The intervals $(-\infty, T_L)$ and (T_U, ∞) are designated "confidence" regions. Interval (T_L, T_U) is designated "no confidence" region.

The N-P test utilizing all the r_is will have the form

$$\sum_{i=1}^{N} r_i > t_b. \tag{24}$$

To achieve a false alarm P_F^b, a threshold of

$$t_b = \sqrt{N}\, Q^{-1}(P_F^b) \tag{25}$$

is needed at the fusion center, where $Q(\ ) = 1 - \Phi(\ )$, with $\Phi(\ )$ the cumulative distribution function (cdf) of the standard normal, and Q^{-1} is the inverse function of Q. Moreover,

$$P_D^b = Q\left(\frac{t_b - NS}{\sqrt{N}}\right). \tag{26}$$

To obtain a $P_F = 0.05$ and $P_D = 0.95$ at each sensor, a signal satisfying $t_i = Q^{-1}(0.05)$ is required, from which $t_i = 1.64$, and $0.05 = 1 - Q(t_i - S)$ from which $S = 3.29$.

Consider achieving a $P_F^b = 0.001$ at the fusion center with the four sensors. This requires a threshold $t_b = 2\,Q^{-1}(0.001) = 6.18$, from which $P_D^b = 0.9998$ (see (25) and (26)).

This example shows that the best decentralized fusion scheme achieves a $(P_F^f, P_D^f) = (0.014, 0.9995)$, whereas the best centralized fusion scheme achieves a $(P_F^b, P_D^b) = (0.001, 0.9998)$ for the same sensors. Clearly the loss in power associated with transmitting highly condensed information from the sensors to the fusion center is causing the degradation in the performance of the fusion scheme. As a compromise, a multibit information could be transmitted to the fusion center containing quality information related to the degree of confidence that a sensor has about its decision along with the decision itself. This situation is examined in Section III.

Table I gives the different N-P test thresholds that the fusion center can operate so that condition (15) is satisfied. The thresholds were found using the interactive fusion algorithm (IFA) that we developed (see the Appendix).

TABLE I

	Decision Fusion	5	Sensor System
	Sensors PF	Equal x	Unequal _
	Sensors PD	Equal x	Unequal _

Threshold @ Fusion Center	Probability of Detection @ Fusion Center	Probability of False Alarm @ Fusion Center
PDMAX = .95000		PFMIN = .50000E-01
t^*	PD	PF
6859.0	.977407	.300000E-04
19.000	.998842	.115812E-02
.52631E-01	.999970	.225925E-01
	1 SENSOR OFF	
PDMAX = .95000		PFMIN = .50000E-01
t^*	PD	PF
361.00	.985981	.481250E-03
1.0000	.999519	.140187E-01
	2 SENSORS OFF	
PDMAX = .95000		PFMIN = .50000E-01
t^*	PD	PF
19.000	.992750	.725000E-02

B. Disimilar Sensors

Case 1. All the sensors operate at the same probability of false alarm level P_F, but different levels of probability of detection from each other, i.e., $P_{D_i} \neq P_{D_j}$, $i \neq j$. Without loss of generality we assume the ranking $P_{D_1} > P_{D_2} > \cdots > P_{D_N}$, from which the following ordering in the abscissae of the conditional distribution of the individual LRs results:

$$\frac{1-P_{D_1}}{1-P_F} < \frac{1-P_{D_2}}{1-P_F} < \cdots < \frac{1-P_{D_N}}{1-P_F} < \frac{P_{D_N}}{P_F} < \cdots < \frac{P_{D_1}}{P_F}.$$

The conditional distribution of the compound LR at the fusion center is obtained by convolving the individual distributions, using the IFA. Convolution of the distributions $P(\log \Lambda(u_i)|H_k)$ corresponds to linear shifts of their logarithmic abscissae, which is translated into addition of logarithms. Hence, the distribution of the LR $P(\Lambda(u)|H_k)$ at the fusion center can be obtained directly by multiplication of the abscissae of the $P(\Lambda(u)_i)|H_k)$. Hence the point of the distribution $P(\Lambda(u)|H_k)$ which is closest to the origin has abscissa $\frac{(1-P_{D_1})\cdots(1-P_{D_N})}{(1-P_F)^N}$ and ordinate $(1-P_{D_1})\cdots(1-P_{D_N})$ under H_1 or $(1-P_F)^N$ under H_0. On the other hand, the point farthest apart from the origin has abscissa $\frac{P_{D_1}P_{D_2}\cdots P_{D_N}}{P_F^N}$ and ordinate $P_{D_1}\cdots P_{D_N}$ under H_1 or P_F^N under H_0. In between these two extreme points, the abscissae of the distribution of the compound LR have the form $\prod_{i\in S}\frac{P_{D_i}}{P_F}\prod_{i\in\bar S}\frac{P_{D_i}}{1-P_F}$ where S is a subset of integers from $\{1, 2, \ldots, N\}$ and $\bar S$ its complement with respect to this set. The corresponding ordinates are $\prod_{i\in S} P_{D_i}\prod_{i\in\bar S}(1-P_{D_i})$ under H_1 or $P_F^{|S|}(1-P_F)^{|\bar S|}$ under H_0, where $|\Omega|$ designates the cardinality of the set Ω. Once the distribution of the compound LR is determined, the threshold at the fusion center can be determined to satisfy a given probability of false alarm P_F^f from which the probability of detection P_D^f is determined. At the fusion center we want to set-up the threshold so that $P_F^f \leq P_F$ while $P_D^f > \max_i\{P_{D_i}\}$. This is achieved by the IFA as the following example illustrates.

Consider a five-sensor system. All the sensors operate at the same level $P_F = 0.05$. However, due to different noise environments or quality of the sensors, they yield different P_Ds as Table II indicates.

TABLE II
Probability Of Detection At The Individual Sensors For The Same Probability Of False Alarm In A Five Sensor System

i	1	2	3	4	5
P_D	0.95	0.94	0.93	0.92	0.91

Table III summarizes all the choices of thresholds at the fusion center that satisfy condition (15) as given by the IFA. A significant improvement in the system performance is achieved by fusing the individual decisions.

Case 2. The different sensors operate at different probabilities of false alarm and probabilities of detection, i.e., $P_{F_i} \neq P_{F_j}$ and $P_{D_i} \neq P_{D_j}$, $i \neq j$. The distribution of the cumulative LR of the fusion center is obtained numerically as in case 2, and the threshold t_f^* is found to satisfy a given P_F^f. Ideally, the threshold t_f^* must be chosen so that condition (15) is satisfied. However, this may not always be feasible. The following examples illustrate the procedure.

We consider three different systems with five, four, and three sensors. Each system results by eliminating the sensor with the lowest P_D from the system that has one more sensor. For the five-sensor system, the (P_D, P_F) of the sensors are given in Table IV.

Table V summarizes the results as obtained by IFA

TABLE III

Decision Fusion	5	Sensor System
Sensors PF	Equal x	Unequal _
Sensors PD	Equal _	Unequal x

Threshold @ Fusion Center	Probability of Detection @ Fusion Center	Probability of False Alarm @ Fusion Center
PDMAX = 95000		PFMIN = 50000E-01
t*	PD	PF
6163 2	957817	300000E-04
53 004	963797	142812E-03
45 880	968973	255625E-03
40 339	973523	368437E-03
38 907	977913	481250E-03
34 208	981772	594062E-03
32.081	985391	706874E-03
29 610	.988731	.819687E-03
28.207	.991913	932499E-03
24 416	994668	.104531E-02
20 705	997003	115812E-02
20998	997454	330156E-02
17806	.997835	544500E-02
15413	998165	.758843E-02
.14683	.998480	973187E-02
13552	.998771	.118753E-01
12709	999043	.140187E-01
11174	.999282	.161622E-01
.10778	.999513	.183056E-01
94760E-01	.999717	.204490E-01
.82023E-01	.999892	.225925E-01

TABLE V

Decision Fusion	5	Sensor System
Sensors PF	Equal _	Unequal x
Sensors PD	Equal _	Unequal x

Threshold @ Fusion Center	Probability of Detection @ Fusion Center	Probability of False Alarm @ Fusion Center
PDMAX = 95000		PFMIN = 10000E-01
t*	PD	PF
57882.	957817	269200E-05
426.86	960153	816400E-05
373 63	962908	155360E-04
358.72	966248	248480E-04
284.83	969430	360200E-04
273.46	.973289	501320E-04
239 36	977840	691439E-04
160.34	981459	917159E-04
153 94	985848	120228E-03
134 74	.991024	158640E-03
102.72	997003	216852E-03
99369	.997179	.393780E-03
75752	.997382	661908E-03
66305	.997622	102314E-02
.63660	.997912	147942E-02
42643	.998143	.202115E-02
.37325	.998416	.275098E-02
35836	.998746	367287E-02
28454	.999061	477889E-02
27319	.999442	.617598E-02
.23912	999892	.805816E-02

1 SENSOR OFF

PDMAX = 95000		PFMIN = 20000E-01
t*	PD	PF
1129 9	.976981	150400E-03
4.6908	.979548	697600E-03
4 1058	.982575	.143480E-02
3 9420	.986246	236600E-02
3 1300	.989742	348320E-02
3 0051	.993983	489440E-02
2.6303	.998984	679560E-02

2 SENSORS OFF

PDMAX = .95000		PFMIN = 30000E-01
t*	PD	PF
32.222	.989720	458000E-02

TABLE IV
Probability Of False Alarm And Detection For A Five-Sensor System With Disimilar Sensors

i	1	2	3	4	5
P_F	0.05	0.04	0.03	0.02	0.01
P_D	0.95	0.94	0.93	0.92	0.91

In all cases, a significant improvement in the performance of the system is achieved from fusing the decisions.

III. TRANSMISSION OF DECISIONS ALONG WITH QUALITY INFORMATION

Consider the case where the jth sensor transmits quality information bits to the fusion center about its decision along with the decision itself. The simplest case corresponds to the transmission of binary {0, 1} quality information indicating the degree of confidence that the sensor has on the decision that it transmits. Under the scenario, a bit one indicates "confidence", whereas a bit zero indicates "no confidence". Fig. 5 illustrates how the binary quality bit c is defined. A strip (T_L, T_U) about the threshold T of an individual sensor is designated as region of no confidence and the bit $c = 0$ is transmitted along with the decision when the observation r falls into this region. The two regions forming the compliment of the (T_L, T_u) region are considered confidence regions and the bit $c = 1$ is transmitted along with the decision when the observations fall into one of the two regions.

The joint probability distribution of (u, c) (skipping the sensor index for simplicity) can be easily obtained from

$$P(u, c|H_k) = P(c|u, H_k)\, P(u|H_k), \qquad k = 0, 1 \qquad (27)$$

where $P(u|H_k)$, $u = \pm 1$ and $k = 0, 1$ is specified by P_F and P_D, and referring to Fig. 5.

$$P(c|u,\mathrm{H}_k) = \begin{cases} P(c=1|u=1,\mathrm{H}_k) = \int_{IV} dP(r|\mathrm{H}_k) / \int_{III \cup IV} dP(r|\mathrm{H}_k) = C_{11}^k \\ P(c=0|u=1,\mathrm{H}_k) = \int_{III} dP(r|\mathrm{H}_k) / \int_{III \cup IV} dP(r|\mathrm{H}_k) = C_{01}^k \\ P(c=0|u=-1,\mathrm{H}_k) = \int_{II} dP(r|\mathrm{H}_k) / \int_{I \cup II} dP(r|\mathrm{H}_k) = C_{00}^k \\ P(c=1|u=-1,\mathrm{H}_k) = \int_{I} dP(r|\mathrm{H}_k) / \int_{I \cup II} dP(r|\mathrm{H}_k) = C_{10}^k \end{cases} \qquad (28)$$

for $k = 0, 1$.

Hence, for every sensor

$$P(u=i,\ c=j|\mathrm{H}_k) = C_{ji}^k\, P(u=i|\mathrm{H}_k), \qquad i = -1, 1, \text{ and } j = 0, 1 \qquad (29)$$

and

$$\Lambda(u=i,\ c=j) = \frac{P(u=i,\ c=j|\mathrm{H}_1)}{P(u=i,\ c=j|\mathrm{H}_0)} = \frac{C_{ji}^1\, P(u=i|\mathrm{H}_1)}{C_{ji}^0\, P(u=i|\mathrm{H}_0)}, \qquad i = -1, 1, \text{ and } j = 0, 1. \qquad (30)$$

Combining (6) and (22) we obtain

$$\begin{aligned} P(\Lambda(u,c)|\mathrm{H}_1) = {} & C_{11}^1\, P_D\, \delta\left(\Lambda(u,c) - \frac{C_{11}^1\, P_D}{C_{11}^0\, P_F}\right) \\ & + C_{01}^1\, P_D\, \delta\left(\Lambda(u,c) - \frac{C_{01}^1\, P_D}{C_{01}^0\, P_F}\right) \\ & + C_{00}^1 (1 - P_D)\, \delta\left(\Lambda(u,c) - \frac{C_{00}^1\, (1 - P_D)}{C_{00}^0\, (1 - P_F)}\right) \\ & + C_{10}^1 (1 - P_D)\, \delta\left(\Lambda(u,c) - \frac{C_{10}^1\, (1 - P_D)}{C_{10}^1\, (1 - P_F)}\right) \end{aligned} \qquad (31)$$

Similarly, $P(\Lambda(u,c)|\mathrm{H}_0)$ is obtained from (29) by substituting P_D with P_F in the product-weights of the delta functions. Therefore, the probability distribution of the LR at the fusion center is given by the convolution

$$P(\log \Lambda(u,c)|\mathrm{H}_k) = P(\log \Lambda(u_1,c_1)|\mathrm{H}_k) * \cdots * P(\log \Lambda(u_N,c_N)|\mathrm{H}_k). \qquad (32)$$

In the case where all the sensors operate at the same level (P_F, P_D) the mathematics simplify somewhat, since

$P(\Lambda(u,c)|\mathrm{H}_1) = \Pr\{k$ out of N decisions favor H_1 and, n out of these k decisions have confidence index 1 and, m out of the $N-k$ decisions that favor H_0 have confidence index $1|\mathrm{H}_1\}$

$$= \binom{k}{n} [C_{11}^1]^n [1 - C_{11}^1]^{k-n} \binom{N-k}{m} \cdot [C_{10}^1]^m [1 - C_{10}^1]^{N-k-m} \cdot \binom{N}{k} P_D^k (1 - P_D)^{N-k} \qquad (33)$$

Similarly,

$$P(\Lambda(u,c)|\mathrm{H}_0) = \binom{k}{n} [C_{11}^0]^n [1 - C_{11}^0]^{k-n} \cdot \binom{N-k}{m} [C_{10}^0]^m [1 - C_{10}^0]^{N-k-m} \cdot \binom{N}{k} P_F^k (1 - P_F)^{N-k} \qquad (34)$$

from which

$$P_F^f = \sum_{k=t_1^*}^{N} \sum_{n=t_2^*}^{k} \sum_{m=t_3^*}^{N-k} \left[\binom{k}{n} [C_{11}^0]^n \cdot [1 - C_{11}^0]^{k-n} \binom{N-k}{m} [C_{10}^0]^m \cdot [1 - C_{10}^0]^{N-k-m} \binom{N}{k} P_F^k (1 - P_F)^{N-k} \right]. \qquad (35)$$

The P_D^f is obtained by an expression similar to (35) with P_D in place of P_F and the index 1 instead of 0 above C_{ij}. The thresholds t_1^*, t_2^*, and t_3^* are to be determined to satisfy a given probability of false alarm at the fusion center. Notice that more than one set of thresholds can yield the same P_F^f. Clearly, the set that results in the highest P_D^f must be selected.

From (35) it can be seen that a superior performance in regards to (P_F^f, P_D^f) can be achieved when quality information is transmitted along with the decisions. The improvement in performance of the fusion center when quality information bits are transmitted comes from the fact that the summation over $P(\Lambda(u,c)|\mathrm{H}_k)$ can be made finer with the three different thresholds. To show that, consider the example of Section IIA. In this example four similar sensors $N = 4$, operate at $P_F = 0.05$ and $P_D = 0.95$ from received data $r_i \sim N(0, 1)$ under H_0 and $r_i \sim N(S = 3.29, 1)$ under H_1. The threshold at each sensor is set to $t_i = 1.64$ to satisfy P_F. Using Fig. 5 and the previous equations, we obtain for $t_{L,i} = 0.8 t_i = 1.312$ and $t_{u,i} = 1.2$, and $t_i = 1.968$ the C_{ij}^ks that are given in Table VI.

Using the IFA, it follows that there is a choice of 33 different thresholds that the fusion center can operate so that (15) is satisfied as shown in Table VII. It can be seen from this table that there is a significant improvement in the performance of the overall system

TABLE VI

Quality Bit Coefficients For Gaussian Distributed Data

H_k / C_{ij}	H_1	H_0
C_{11}	0 948	0 46
C_{01}	0 052	0 54
C_{00}	0.52	0 047
C_{10}	0 48	0 953

TABLE VII

Decision Fusion 4 Sensor System with Quality Bits
Sensors PF Equal x Unequal _
Sensors PD Equal x Unequal _

Threshold @ Fusion Center	Probability of Detection @ Fusion Center	Probability of False Alarm @ Fusion Center
PDMAX = 95000		PFMIN = 50000E-01
t*	PD	PF
62318.	956002	175551E-05
20357	960940	.199808E-05
9390.7	961918	210220E-05
2988 7	963462	.261876E-05
2911.9	980782	856706E-05
951.21	981595	942131E-05
926 74	990711	.192580E-04
438.79	990738	.193191E-04
302.74	990880	197900E-04
139 65	990937	201943E-04
136.06	.992362	306685E-04
44 446	992406	.316713E-04
43 303	993906	663133E-04
42 189	998114	.166041E-03
20 503	998114	166055E-03
14 146	998129	.167161E-03
13 782	998524	195805E-03
6.5253	998525	.195924E-03
6.3575	998577	204121E-03
4 5021	998579	204578E-03
2.0768	998580	.204970E-03
2.0234	998662	.245637E-03
1 9713	999354	.596850E-03
66097	.999355	.597499E-03
64397	999398	.664750E-03
.62741	999762	.124555E-02
.29706	.999763	124796E-02
21036	.999763	124850E-02
.20495	999771	.128557E-02
.94544E-01	999772	.130148E-02
92113E-01	999810	.171378E-02
.66951E-01	.999810	.171395E-02
30090E-01	.999811	.175343E-02
.29316E-01	.999851	.311705E-02

compared with the individual sensors and the fusion system without quality information. For a comparable $P^f_D = 0.9998$, the $P^f_F = 0.0013$ when quality bit information is transmitted as opposed to $(P^f_F, P^f_D) = (0.014, 0.9995)$ without quality information. The performance of the fusion center when one quality information bit is transmitted approaches that of the best centralized N-P test, as Table VIII suggests. It is

TABLE VIII

Comparative Results From 3 Different Fusion Systems With Four ($N = 4$) Sensors. All Operating At Level $(P_F, P_D) = (0.05, 0.95)$ When The Individual Sensors Transmit

	P^f_F	P^f_D
Only decisions	0 014	0 9995
Decision with one quality bit	0 0013	0 9998
Raw data (Best centralized N-P test)	0 001	0 9998

interesting to notice that fusion of the decisions improves the performance of the overall system even in the case of two sensors when quality information bits are transmitted along with the decisions, as Table IX indicates. Table X shows the performance of a three sensor system with quality bits.

TABLE IX

Decision Fusion . 2 Sensor System with Quality Bits
Sensors PF . Equal x Unequal _
Sensors PD Equal x Unequal _

Threshold @ Fusion Center	Probability of Detection @ Fusion Center	Probability of False Alarm @ Fusion Center
PDMAX = 95000		PFMIN = 50000E-01
t*	PD	PF
1.0654	951900	696499E-02
1.0380	995129	486111E-01

IV. CONCLUSIONS

The problem of fusing decisions from N independent sensors in a fusion center was considered. We assumed *that each sensor transmits its decision to the fusion* center. The decision of each individual sensor is based on the N-P test. The fusion center formulates the LR using all the received decisions and decides on which hypothesis is true using the N-P test also. The pdf of the

TABLE X

Decision Fusion : 3 Sensor System with Quality Bits
Sensors PF : Equal x Unequal _
Sensors PD : Equal x Unequal _

Threshold @ Fusion Center	Probability of Detection @ Fusion Center	Probability of False Alarm @ Fusion Center
PDMAX = .95000		PFMIN = 50000E-01
t*	PD	PF
40.645	.985857	177933E-02
13.277	.987683	191689E-02
6.1248	.987804	193657E-02
1.9493	.987994	203422E-02
1.8992	994400	540756E-02
.62039	.994500	556904E-02
.60444	997872	111475E
.19745	997890	112365E-01
.88740E-01	.998065	132165E
.28243E-01	998250	197652E

log LR at the fusion center was obtained as the convolution of the pdfs of the log LRs of the individual sensors. Once the pdf of the LR is obtained, the threshold at the fusion center is determined by a desired probability of false alarm.

For a fusion system with three or more sensors, all the sensors operating at the same (P_F, P_D) level, it was proved that if the N-P test is used to fuse the decisions, the probability of detection at the fusion center exceeds that of the individual sensor for the same probability of false alarm. However, if the sensors operate at arbitrary (P_F, P_D) levels, no general assessment can be made about the performance of the fusion center since the performance depends on how far the operating points of the sensors are from each other.

The problem of decision fusion when the sensors transmit quality information bits indicating their confidence on the decisions was also considered and the N-P test at the fusion center was derived. Several numerical examples showed that use of quality information can improve the performance of the fusion center considerably.

An IFA was developed to solve the fusion problem numerically. Once one of the three parameters (threshold, probability of false alarm, or probability of detection) is specified, the IFA determines the other two, given the probabilities of false alarm and detection of each individual sensor.

APPENDIX

The IFA receives as data the number of sensors, their (P_F, P_D) levels, and the C_{ij}^k quality information parameters if the sensors transmit quality information bits along with their decisions. It then computes the LR pdf at the fusion center conditioned on each hypothesis. After it computes the pdf, it asks the user which option he/she prefers. The alternative options are the following.

1) Display of the entire pdf.
2) Threshold computation for a given P'_F and display of the corresponding P'_D.
3) Determination of the thresholds that satisfy (15).
4) Threshold computation for a given P'_D and display of the corresponding P'_F.
5) Elimination of one or more sensors and repetition of the algorithm.
6) Quit.

REFERENCES

[1] Tenney, R.R., and Sandell, N.R., Jr (1981)
Detection with distributed sensors.
IEEE Transactions on Aerospace and Electronic Systems, *AES-17* (July 1981), 501–510.

[2] Sadjadi, F.A. (1986)
Hypothesis testing in a distributed environment
IEEE Transactions on Aerospace and Electronic Systems, *AES-22* 2 (Mar. 1986), 134–137

[3] Teneketzis, D., and Varaiya, P. (1984)
The decentralized quickest detection problem.
IEEE Transactions on Automatic Control, *AC-29*, 7 (July 1984), 641–644

[4] Teneketzis, D (1982)
The decentralized Wald problem
In *Proceedings of the IEEE 1982 International Large-Scale Systems Symposium*, Virginia Beach, Virginia, Oct 1982, pp 423–430.

[5] Tsitsiklis, J., and Athans, M (1985)
On the complexity of distributed decision problems
IEEE Transactions on Automatic Control, *AC-30*, 5 (May 1985), 440–446

[6] Chair, Z., and Varshney, P.K. (1986)
Optimal data fusion in multiple sensor detection systems
IEEE Transactions on Aerospace and Electronic Systems, *AES-22*, 1 (Jan. 1986), 98–101

Stelios C.A. Thomopoulos (S'82, M'83) was born in Athens, Greece, on June 15, 1955. He received his Diploma of Electrical & Mechanical Engineering from the National Technical University of Athens in 1978, and both his M.S. and Ph.D. in electrical engineering from the State University of New York at Buffalo in 1981 and 1983, respectively.

Since 1983, he has been an Assistant Professor in the Department of Electrical Engineering at Southern Illinois University, Carbondale, Il. His areas of interest include data networks, distributed processing, robotics, and artificial intelligence. Dr Thomopoulos is a member and licensed Engineer of the Technical Chamber of Greece

R. Viswanathan (S'81, M'83) received the B.E. (Hons.) degree in electronics and communication engineering from the University of Madras, India, in 1975, the M.E. degree with distinction in electrical communication engineering from the Indian Institute of Science, Bangalore, India in 1977, and the Ph.D. degree in electrical engineering from Southern Methodist University, Dallas, Tex. in 1983.

From 1977 until 1980 he worked as a Deputy Engineer in the Digital Communication Department of the Research and Development Division of Bharat Electronics Limited, Bangalore, India. Since 1983 he has been an Assistant Professor in the Electrical Engineering Department at Southern Illinois University at Carbondale, Ill. His research interests include statistical theory of communication, spread-spectrum systems, and mobile radio systems. Dr. Viswanathan is a member of Tau Beta Pi

Dimitrios Bougoulias was born in Thessaloniki, Greece, on Mar. 5, 1958. He received the B.S. degree in physics from the National University of Greece, Athens in 1982, and the M.S. degree in electrical engineering from Southern Illinois University, Carbondale, Il. in 1986.

He is currently studying towards his Ph.D. degree in engineering science at Southern Illinois University.

Correspondence

Optimal Distributed Decision Fusion

The problem of decision fusion in distributed sensor systems is considered. Distributed sensors pass their decisions about the same hypotheses to a fusion center that combines them into a final decision. Assuming that the sensor decisions are independent from each other conditioned on each hypothesis, we provide a general proof that the optimal decision scheme that maximizes the probability of detection at the fusion for fixed false alarm probability consists of a Neyman-Pearson test (or a randomized N-P test) at the fusion and likelihood-ratio tests at the sensors.

I. INTRODUCTION

Systems of distributed sensors monitoring a common volume and passing their decisions into a centralized fusion center which further combines them into a final decision have been receiving a lot of attention in recent years [1]. Such systems are expected to increase the reliability of the detection and be fairly immune to noise interference and to failures. In a number of papers the problem of optimally fusing the decisions from a number of sensors has been considered. Tenney and Sandell [2] have considered the Bayesian detection problem with distributed sensors without considering the design of data fusion algorithms. Sadjadi [3] has considered the problem of hypothesis testing in a distributed environment and has provided a solution in terms of a number of coupled nonlinear equations. The decentralized sequential detection problem has been investigated in [4, 5]. In [6] it was shown that the solution of distributed detection problems is nonpolynomial complete. Chair and Varshney [7] have solved the problem of data fusion when the a-priori probabilities of the tested hypotheses are known and the likelihood-ratio (L-R) test can be implemented at the receiver. Thomopoulos, Viswanathan, and Bougoulias [8, 9] have derived the optimal fusion rule for unknown a-priori probabilities in terms of the Neyman-Pearson (N-P) test.

For the "parallel" sensor topology of Fig. 1, Srinivasan [10] has shown that the globally optimal solution to the fusion problem that maximizes the probability of detection for fixed probability of false alarm when sensors transmit independent, binary decisions to the fusion center, consists of L-R tests

Manuscript received March 31, 1987; revised January 10, 1989.

IEEE Log No. 30105.

This work was supported by the SDIO/IST and managed by the Office of Naval Research under Contract N00014-86-K-0515.

0018-9251/89/0900-0761 $1.00 © 1989 IEEE

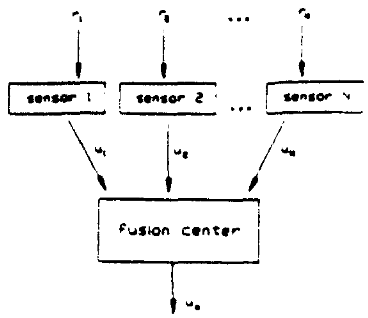


Fig. 1. Distributed sensor fusion. Parallel topology.

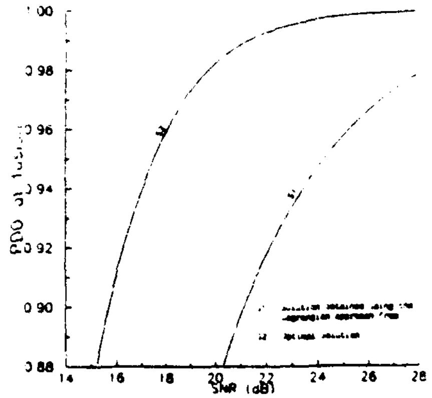


Fig. 2. Example of singularity of Lagrangian approach used in [10] for decision fusion. Three identical sensors in slow-fading Rayleigh channel. Paradigm taken from [11].

at all sensors and a N-P test at the fusion center. This test will be referred to as N-P/L-R hereafter. The proof of the optimality of the N-P/L-R test in [10] is based on the (first-order) Lagrange multipliers methods which does not always yield the optimal solution as it is shown by example in [11]. For the paradigm in [11], the Lagrangian approach fails to yield to optimal solution. Instead, it yields a solution which is by far inferior to the optimal solution, see Fig. 2. A detailed description and analysis of this singular case is given in [11, 12]. A theoretical explanation of the failure of the Lagrange multipliers method can be found in [13, ch. 5, and 14, 15].

In general, if the optimal solution lies on the boundary of the domain of x (as in the decision fusion paradigm in [11]), the Lagrangian formulation fails to guarantee the convexity of the objective function, and thus, the optimality of the solution obtained using the Lagrange multipliers method. In that sense, the proof of optimality of the N-P/L-R test for the parallel sensor topology in [10], which is based on a Lagrangian formulation, is incomplete. We give a complete proof of the optimality of the N-P/L-R test for the distributed decision fusion problem that does not depend on the Lagrangian formulation.

II. OPTIMALITY OF N-P/L-R TEST IN DISTRIBUTED DECISION FUSION

A number of sensors N receive data from a common volume. Sensor k receives data r_k and generates the first stage decision u_k, $k = 1,2,\ldots,N$. The decisions are subsequently transmitted to the fusion center where they are combined into a final decision u_0 about which of the hypotheses is true, Fig. 1. Assuming binary hypothesis testing for simplicity, we use $u_i = 1$ or 0 to designate that sensor i favors hypotheses H_1 or H_0, respectively. In order to derive the globally optimal fusion rule we assume that the received data r_k at the N sensors are statistically independent, conditioned on each hypothesis. This implies that the received decisions at the fusion center are independent conditioned on each hypothesis. Improvement in the performance of conventional diversity schemes is based on the validity of this assumption [16]. Given a desired level of probability of false alarm at the fusion center, $P_{F_0} = \alpha_0$, the test that maximizes the probability of detection P_{D_0} (thus, minimizes the probability of miss $P_{M_0} = 1 - P_{D_0}$) is the N-P test [17, 18]. Because of the comparison to a threshold this test is referred to as a threshold optimal test.

Next, we prove that the optimal solution to the fusion problem involves an N-P test at the fusion center and L-R tests at the sensors.

Let $d(u_1,u_2,\ldots,u_N)$ be the (binary) decision function (rule) at the fusion. Since $d(u_1,u_2,\ldots,u_N)$ is either 0 or 1, and all the possible combinations of decisions $\{u_1,u_2,\ldots,u_N\}$ that the fusion center can receive from the N sensors is 2^N, the set of all possible decision functions contain 2^{2^N} d functions. However, not all these functions d can be threshold optimal as the next Lemma states.

LEMMA 1. *Let the sensors individual decisions u_k be independent from each other conditioned on each hypothesis. Let $P_{F_i} = P(u_i = 1 \mid H_0)$ be the false alarm probability and $P_{D_i} = P(u_i = 1 \mid H_1)$ be the probability of detection at the ith sensors. Assuming, without loss of generality, that for every sensor $P_{D_i} \geq P_{F_i}$, a necessary condition for a fusion function $d(u_1,u_2,\ldots,u_N)$ to be threshold optimal is*

$$d(A_k, U - A_k) = 1 \Rightarrow d(A_n, U - A_n) = 1 \quad \text{if} \quad A_n > A_k \qquad (1)$$

where $U = \{u_1,u_2,\ldots,u_N\}$ denotes the set of the peripheral sensor decisions, A_k is a set of decisions with k sensors favoring hypothesis H_1 *(whereas the complement set of decisions $U - A_k$ favors hypothesis* H_0), *and A_n is any set that contains the decisions from these k sensors. [The symbol ">" is used to indicate "greater than" in the standard multidimensional coordinate-wise sense, i.e., $A_n > A_k$ if and only if $u_{n_i} \geq u_{k_i} \forall i$, $i = 1,2,\ldots,N$, with at least one holding as* *a strict inequality, where $u_{n_i}(u_{k_i})$ indicates the decision of the same ith sensor in the $A_n(A_k)$ decision set.*]

PROOF. Let $P_{F_i} = P(u_i = 1 \mid H_0)$ be the false alarm probability and $P_{D_i} = P(u_i = 1 \mid H_1)$ be the probability of detection at the ith sensors. $d(A_k, U - A_k) = 1$ implies that the likelihood ratio

$$\frac{p(A_k, U - A_k \mid H_1)}{p(A_k, U - A_k \mid H_0)} = \frac{p(A_k \mid H_1)p(U - A_k \mid H_1)}{p(A_k \mid H_0)p(U - A_k \mid H_0)} > \lambda_0 \quad (2)$$

which in turn implies that, for $A_n > A_k$,

$$\frac{p(A_n, U - A_n \mid H_1)}{p(A_n, U - A_n \mid H_0)} = \frac{p(A_k \mid H_1)p(A_n - A_k \mid H_1)p(U - A_n \mid H_1)}{p(A_k \mid H_0)p(A_n - A_k \mid H_0)p(U - A_n \mid H_0)} \geq \frac{p(A_k \mid H_1)p(U - A_k \mid H_1)}{p(A_k \mid H_0)p(U - A_k \mid H_0)} > \lambda_0 \quad (3)$$

since, under the assumption that $P_{D_i} \geq P_{F_i}$ for every sensor i,

$$\frac{P(u_i = 1 \mid H_1)}{P(u_i = 1 \mid H_0)} = \frac{P_{D_i}}{P_{F_i}} \geq \frac{P(u_i = 0 \mid H_1)}{P(u_i = 0 \mid H_0)} = \frac{1 - P_{D_i}}{1 - P_{F_i}}. \quad (4)$$

From (3), it follows that $d(A_n, U - A_n) = 1$.

REMARK 1. Functions that do not satisfy (2) cannot lead to the set of optimal thresholds. A function d that satisfies Lemma 1, is called a monotone increasing function in the context of switching and automata theory, Table I, [19].

REMARK 2. If $P_{D_i} = P_{F_i}$ for all sensors, the L-R at the fusion is degenerated to one, identically for any combination of the peripheral decisions [9]. Hence, for any likelihood test, the false alarm probability P_{F_0} and the detection probability P_{D_0} at the fusion are either a) both one, if the threshold is less or equal to one, or b) both zero, if the threshold is greater than one. In the first case, the fusion rule always favors hypothesis one, independent of the combination of sensor decisions, i.e., $d(U) = 1$ for all Us, which is a monotone increasing function satisfying Lemma 1. In the second case, the fusion rule always favors hypothesis zero, independent of the combination of sensor decisions, i.e., $d(U) = 0$ for all Us, which is a monotone increasing function satisfying Lemma 1.

REMARK 3. If $P_{D_i} \leq P_{F_i}$ for all sensors, the inequality in (3) is reversed, and Lemma 1 still holds with all threshold optimal decisions at the fusion being monotonically increasing functions of the sensor decisions.

REMARK 4. If for some sensors $P_{D_i} \geq P_{F_i}$ while for some others $P_{D_j} \leq P_{F_j}$, Lemma 1 does not hold. However, this is an uninteresting case, for if we wish to maximize the detection probability at the fusion, we would either ignore the sensors for which $P_{D_j} \leq P_{F_j}$, or, randomize their decisions by flipping coins and deciding with probability 1/2 for either one of the two hypotheses.

LEMMA 2. *For any fixed threshold λ_0 and any fixed monotonic function $t(u_1, u_2, \ldots, u_N)$, P_{D_0} is an increasing function of the P_{D_i}s, $i = 1, 2, \ldots, N$.*

PROOF. The decision function that corresponds to the likelihood test at the fusion is contained in the set of monotone functions of N variables. Consider one such monotone increasing decision function $d(u_1, u_2, \ldots, u_N)$. The function d, when expressed in sum of product form in the Boolean sense [19], contains only some of the literals $u_1, \ldots, u_N$ in the uncomplemented form and none of the complemented variables $(\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_N)$. Since the random variables $u_1, u_2, \ldots, u_N$ are statistically independent, it is possible to compute P_{D_0} knowing the P_{D_i}s [9, eq. (20)–(22)]. Taking partial derivatives of the P_{D_0} w.r.t. P_{D_i}s, one obtains that $(\partial P_{D_0}/\partial P_{D_i}) > 0\ \forall i$, i.e., the desired result. (As an illustration, consider the function $d(u_1, u_2, u_3) = u_1 + u_2 u_3$. For this function $P_{D_0} = P_{D_1} + P_{D_2}P_{D_3} - P_{D_1}(P_{D_2}P_{D_3})$, from which, $(\partial P_{D_0}/\partial P_{D_i}) > 0$, $i = 1, 2, 3$.)

THEOREM 1. *Under the assumption of statistical independence of the sensor decisions conditioned on each hypothesis, the optimal decision fusion rule for the parallel sensor topology consists of an N-P test (or, a randomized N-P test) at the fusion and L-R tests at all sensors.*

PROOF. Given the decisions $u_1, u_2, \ldots, u_N$ at the fusion center, the best fusion rule which achieves maximum P_{D_0} for fixed $P_{F_0} = \alpha_0$ is the N-P test (assuming that the false alarm probability α_0 is realizable by an N-P test at the fusion; the randomized case is treated separately afterwards). Call the best test at the fusion center $t(u_1, \ldots, u_N) \gtrless_{H_0}^{H_1} \lambda_0$. From Lemma 1, it follows that the decision function that corresponds to the above test must be one of the monotone increasing functions $d(u_1, u_2, \ldots, u_N)$. Assume that the individual sensors use some test other than the L-R test and are operating with $\{(P_{F_i}, P_{D_i})\ \forall i\}$ such that the condition $P_F = \alpha_0$ is met. From [8, 9] it is seen that P_{F_0} is a function of the P_{F_i}s only, and that P_{D_0} is a function of the P_{D_i}s only. Furthermore, from Lemma 2, P_{D_0} is a monotonic increasing function of the P_Ds. Therefore, the L-R tests at the sensors which operate with $(P_{F_i}^* = P_{F_i}, P_{D_i}^*)$ lead to the best performance at the fusion, since in this case, the achieved $P_{D_0}^*$ is greater than or equal to P_{D_0} that can be achieved with any other test at the sensors.

If the false alarm probability α_0 is not achievable by an N-P test, a randomized N-P maximizes the

[12] Thomopoulos, S. C. A., Viswanathan, R., and Bougoulias, D. K. (1988)
Optimal and suboptimal distributed decision fusion.
22nd Annual Conference on Information Sciences and Systems, Princeton University, NJ, Mar. 16–18, 1988, 886–890.

[13] Hestenes, M. R. (1975)
Optimization Theory: The Finite Dimensional Case.
New York: Wiley, 1975.

[14] Thomopoulos, S. C. A., and Viswanathan, R. (1988)
Optimal and suboptimal distributed decision fusion.
Technical Report, TR-SIU-DEE-87-5, Department of Electrical Engineering, Southern Illinois University, Carbondale, Jan. 1988.

[15] Thomopoulos, S. C. A., Bougoulias, D. K., and Zhang, L. (1988)
Optimal and suboptimal distributed decision fusion.
Presented at SPIE 1988 Technical Symposium on Optics, Electro-Optics, and Sensors, Apr. 4–8, 1988, Orlando, FL.

[16] Viswanathan, R., Thomopoulos, S. C. A., and Tumuluri, R. (1988)
Optimal serial distributed decision fusion.
IEEE Transactions on Aerospace and Electronic Systems, AES-24, 4 (July 1988), 366–376.

[17] Van Trees, H. L. (1968)
Detection Estimation and Modulation Theory, Vol I.
New York: Wiley, 1968.

[18] DiFranco, J. V., and Rubin, W. L. (1980)
Radar Detection.
Dedham, MA: Artech House, 1980.

[19] Harrison, M. A. (1965)
Introduction to Switching and Automata Theory.
New York: McGraw-Hill, 1965.

Optimal Serial Distributed Decision Fusion

R. VISWANATHAN, Member, IEEE
S.C.A. THOMOPOULOS, Member, IEEE
R. TUMULURI
Southern Illinois University at Carbondale

The problem of distributed detection involving N sensors is considered. The configuration of sensors is serial in the sense that the $(j-1)$th sensor passes its decision to the jth sensor and that the jth sensor decides using the decision it receives and its own observation. When each sensor employs the Neyman-Pearson test, the probability of detection is maximized for a given probability of false alarm, at the Nth stage. With two sensors, the serial scheme has a performance better than or equal to the parallel fusion scheme analyzed in the literature. Numerical examples illustrate the global optimization by the selection of operating thresholds at the sensors.

Manuscript received February 2, 1987; revised December 22, 1987.

This work is supported by the SDIO/IST and managed by the Office of Naval Research, Washington, D.C., under Contract N00014-86-K-0515.

Authors' current addresses: R. Viswanathan and S.C.A. Thomopoulos, Department of Electrical Engineering, College of Engineering and Technology, Southern Illinois University at Carbondale, Carbondale, IL 62901-6603; R. Tumuluri, Department of Electrical Engineering, Purdue University, West Lafayette, IN.

0018-9251/88/0700-0366 $1.00 © 1988 IEEE

I. INTRODUCTION

The theory of distributed detection is receiving a lot of attention in the literature [1-10]. Typically, a number of sensors process the data they receive and decide in favor of one of the hypotheses about the origin of the data. In a two-class decision problem, the hypotheses would be signal present (H_1) or the signal absent (H_0). These decisions are then sent to a fusion center where a final decision regarding the presence of the signal is made. This scheme, which can be termed parallel decision making, is shown in Fig. 1. In order to maximize the probability of detection at the fusion center for a fixed probability of false alarm, the tests used at the fusion center and at the sensors must be Neyman-Pearson (N-P) [3, 8]. The above result is based on the assumption that the data at the sensors conditioned on the hypothesis are statistically independent. If the conditional independence is removed, the threshold of the N-P tests become data dependent and does not yield any easy solution for optimization [16].

We consider a serial distributed decision scheme (Fig. 2), (in [4] this is called a tandem network). Though the serial fusion is very sensitive to link failures, its performance analysis is of interest. In [4], the tandem network was analyzed with Baye's cost as the optimality criterion. Though analytical equations are given, no performance analysis for typical channels or comparison of performance with respect to the parallel fusion was provided. Here we aim to fill this gap.

In Section II we derive the relevant equations describing the operation of the serial scheme based on the knowledge that all the sensors employ the N-P test. In Section III we show that the global optimality is guaranteed when each stage employs the N-P test. Section IV examines the conditions under which the performance of the serial scheme is definitely not inferior to the parallel scheme. Some numerical examples are also presented to illustrate the performance.

II. DEVELOPMENT OF KEY EQUATIONS

Consider the serial configuration of distributed sensors shown in Fig. 2. Denote the sensor decisions as $u_1, u_2, \ldots, u_N$. The jth sensor receives the decision u_{j-1} and its own observation Z_j to make its decision u_j. The decision u_N at the Nth sensor is the fused decision about the hypotheses. We assume that the data at the sensors, conditioned on each hypothesis, are statistically independent. This implies that Z_j and u_{j-1} are also conditionally independent. As mentioned earlier, the jth sensor employs an N-P test using the data (Z_j, u_{j-1}). The optimality of this assumption is explored in the next section.

Denoting the distributions of Z_j as $p(Z_j|H_1)$ and $p(Z_j|H_0)$, the likelihood ratio becomes

$$\frac{L(Z_j, u_{j-1}|H_1)}{L(Z_j, u_{j-1}|H_0)}$$

$$= \frac{p(Z_j|H_1)[P_{D,j-1}\delta(u_{j-1}-1) + (1-P_{D,j-1})\delta(u_{j-1})]}{p(Z_j|H_0)[P_{F,j-1}\delta(u_{j-1}-1) + (1-P_{F,j-1})\delta(u_{j-1})]} \quad (1)$$

where

$P_{D,j-1} = \Pr(u_{j-1} = 1|H_1)$

$P_{F,j-1} = \Pr(u_{j-1} = 1|H_0)$

$u_{j-1} = k$ implies that the $(j-1)$th sensor decides H_k, $k = 0, 1$, and $\delta(x)$ is the Kronecker delta function defined as $\delta(x) = \begin{cases} 1 & x = 0 \\ 0 & x \neq 0 \end{cases}$ and $L(\ )$ is the likelihood function [14].

Therefore, the test at the jth sensor is given by

$$\frac{p(Z_j|H_1)}{p(Z_j|H_0)} \frac{P_{D,j-1}}{P_{F,j-1}} \underset{H_0}{\overset{H_1}{\gtrless}} t, \quad \text{if } u_{j-1} = 1$$

$$\frac{p(Z_j|H_1)}{p(Z_j|H_0)} \frac{1 - P_{D,j-1}}{1 - P_{F,j-1}} \underset{H_0}{\overset{H_1}{\gtrless}} t, \quad \text{if } u_{j-1} = 0 \quad (2)$$

where t a threshold to be determined.

Equivalently,

$$\Lambda(Z_j) \underset{H_0}{\overset{H_1}{\gtrless}} \begin{cases} t_{j,1}, & \text{if } u_{j-1} = 1 \\ t_{j,0}, & \text{if } u_{j-1} = 0 \end{cases} \quad (3)$$

where

$$\Lambda(Z_j) = \frac{p(Z_j|H_1)}{p(Z_j|H_0)}$$

and

$$\frac{t_{j,1}}{t_{j,0}} = \frac{P_{F,j-1}}{P_{D,j-1}} \frac{1 - P_{D,j-1}}{1 - P_{F,j-1}}.$$

Many times it is convenient to use the log likelihood ratio, $\ln \Lambda(Z_j) = \Lambda^*(Z_j)$. Hence,

$$\Lambda^*(Z_j) \underset{H_0}{\overset{H_1}{\gtrless}} \begin{cases} t^*_{j,1}, & \text{if } u_{j-1} = 1 \\ t^*_{j,0}, & \text{if } u_{j-1} = 0 \end{cases} \quad (4)$$

and

$$t^*_{j,1} = t^*_{j,0} + \ln\left(\frac{P_{F,j-1}}{1 - P_{F,j-1}} \frac{1 - P_{D,j-1}}{P_{D,j-1}}\right), \quad j = 2, \ldots, N.$$

For the first stage, $t^*_{1,1} = t^*_{1,0}$.

A. False Alarm and Detection Probabilities

At the jth stage, the false alarm probability is giver by

$$P_{F,j} = \Pr(\Lambda^*(Z_j) > t^*_{j,0}|H_0, u_{j-1} = 0) \Pr(u_{j-1} = 0|H_0) + \Pr(\Lambda^*(Z_j) > t^*_{j,1}|H_0, u_{j-1} = 1) \times \Pr(u_{j-1} = 1|H_0). \quad (5)$$

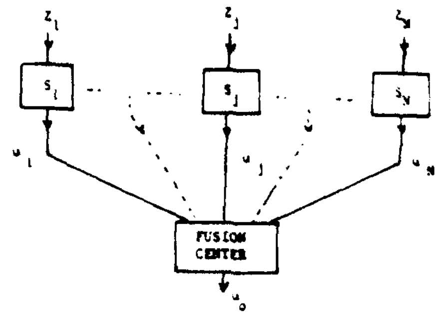


Fig. 1. Parallel decision fusion

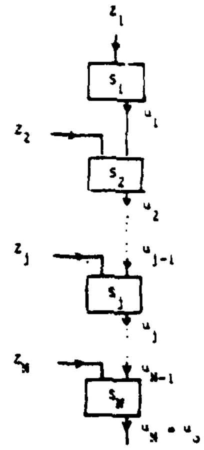


Fig. 2. Serial decision fusion.

Let

$a_j = \Pr(\Lambda^*(Z_j) > t^*_{j,0}|H_0)$

$b_j = \Pr(\Lambda^*(Z_j) > t^*_{j,1}|H_0)$

$c_j = \Pr(\Lambda^*(Z_j) > t^*_{j,0}|H_1)$

$d_j = \Pr(\Lambda^*(Z_j) > t^*_{j,1}|H_1). \quad (6)$

Using (5), (6), and the conditional independence assumption, we have

$$P_{F,j} = a_j(1 - P_{F,j-1}) + b_j P_{F,j-1}. \quad (7)$$

Similarly,

$$P_{D,j} = c_j(1 - P_{D,j-1}) + d_j P_{D,j-1}. \quad (8)$$

Knowing the distribution of the observations Z_j and using (4), (6)–(8), it is possible to compute the $P_{D,j}$s recursively provided the $P_{F,j}$s are specified. If the $P_{F,j}$s are kept the same, the serial configuration exhibits some nice properties [5]. However, for a given $P_{F,N}$ at the Nth stage, this procedure does not guarantee a maximum $P_{D,N}$. In order to globally optimize the performance, that is to maximize $P_{D,N}$ for a given $P_{F,N}$, we need a multidimensional search with respect to the variables $P_{F,j}$s, $j = 1, 2, \ldots, (N-1)$. The results obtained using the numerical search procedure are presented in Section IV.

In Fig. 3 a functionally equivalent form of the serial decision fusion is shown. Each sensor, except the first one, sends two decisions $u_{i,0}$ and $u_{i,1}$ depending on whether the previous sensor decides a 0 or a 1, respectively. These decisions are arrived by using (3). The fusion center uses the decision from the first sensor and sequentially picks the appropriate decisions from the sensors to arrive at the final decision u_0 which is either $u_{N,0}$ or $u_{N,1}$. Performance-wise, the configuration in Fig. 3 is equivalent to the serial scheme. The equivalent configuration does not have the time delay problem associated with the serial configuration. However, both are highly sensitive to link failures.

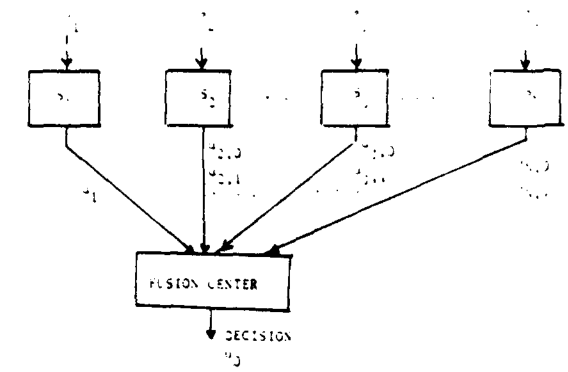


Fig. 3. Functionally equivalent configuration of serial network

III. GLOBAL OPTIMALITY

The global optimization problem is to find the tests at each stage of the serial configuration such that the probability of detection $P_{D,N}$ is maximized for a given $P_{F,N}$. Here, we show that the global optimality is achieved when each sensor employs the N–P test.

THEOREM 1. *Given that the observations at each stage in a serial distributed detection environment with N sensors are independent identically distributed (IID), the probability of detection is maximized for a given probability of false alarm, at the Nth stage, when each stage employs the N–P test.*

PROOF. Consider the last two stages. At the Nth stage, the N–P test using the data (Z_N, u_{N-1}) maximizes $P_{D,N}$ for a fixed $P_{F,N}$ [11, 13]. Let

$$L^* = \ln \frac{p(Z_N, u_{N-1}|H_1)}{p(Z_N, u_{N-1}|H_0)}$$

$$\Lambda^*(Z_N) = \ln \frac{p(Z_N|H_1)}{p(Z_N|H_0)}. \tag{9}$$

Call $\Lambda^*(Z_N)$, $P_{F,N-1}$, and $P_{D,N-1}$ as Λ^*, P_F and P_D, respectively, for simplicity. Then,

$$\Pr(L^* < \lambda|H_1) = P_D \Pr\left(\Lambda^* + \ln\left(\frac{P_D}{P_F}\right) < \lambda \,|\, H_1\right) + (1-P_D)\Pr\left(\Lambda^* + \ln\left(\frac{1-P_D}{1-P_F}\right) < \lambda \,|\, H_1\right) \tag{10}$$

Denote the cumulative distributions and the density functions of Λ^* under H_1 and H_0 as $F_1^*(\ )$, $f_1^*(\ )$ and $F_0^*(\ )$, $f_0^*(\ )$, respectively. Since the left-hand side of (10) is one minus the probability of detection, we have

$$1 - P_{D,N} = P_D F_1^*\left(\lambda - \ln\left(\frac{P_D}{P_F}\right)\right) + (1-P_D) F_1^*\left(\lambda - \ln\left(\frac{1-P_D}{1-P_F}\right)\right) \tag{11}$$

Similarly,

$$1 - P_{F,N} = P_F F_0^*\left(\lambda - \ln\left(\frac{P_D}{P_F}\right)\right) + (1-P_F) F_0^*\left(\lambda - \ln\left(\frac{1-P_D}{1-P_F}\right)\right). \tag{12}$$

We require for a fixed $P_{F,N}$ and for any arbitrary but fixed P_F at the $(N-1)$th stage, the $P_{D,N}$ to be a monotonic increasing function of the P_D at the $(N-1)$th stage. Observe that if the P_D of the $(N-1)$th stage is changed, then the threshold λ at the Nth stage changes in order that $P_{F,N}$ remains fixed. Taking the derivative of (12) w.r.t. P_D and equating the result to zero, we obtain

$$\frac{d\lambda}{dP_D} = \frac{\dfrac{P_F}{P_D} f_0^*(x_1) - \dfrac{1-P_F}{1-P_D} f_0^*(x_2)}{P_F f_0^*(x_1) + (1-P_F) f_0^*(x_2)} \tag{13}$$

where

$$x_1 = \lambda - \ln(P_D/P_F)$$

$$x_2 = \lambda - \ln\left(\frac{1-P_D}{1-P_F}\right).$$

Similarly,

$$\frac{d(1-P_{D,N})}{dP_D} = F_1^*(x_1) - F_1^*(x_2) + \left[P_D f_1^*(x_1)\left(\frac{d\lambda}{dP_D} - \frac{1}{P_D}\right) + (1-P_D) f_1^*(x_2)\left(\frac{d\lambda}{dP_D} + \frac{1}{1-P_D}\right)\right]. \tag{14}$$

A reasonable requirement is $P_D > P_F$. This implies that $F_1^*(x_1) - F_1^*(x_2)$ is less than zero. Hence, a sufficient condition for $\frac{dP_{D,N}}{dP_D} > 0$ is that the term in the brackets in (14) be less than or equal to zero. After some simplification, using (13), we obtain the following sufficiency condition:

$$\frac{\dfrac{f_1^*(x_2)}{f_0^*(x_2)}}{\dfrac{f_1^*(x_1)}{f_0^*(x_1)}} \leq e^{[illegible]} \tag{15}$$

However, from the result that the likelihood ratio of the likelihood ratio is the likelihood ratio itself [11, pp. 46], it follows that (15) is satisfied with equality.

IV. PERFORMANCE ANALYSIS

A. Numerical Results

By using the algorithm developed in Section II, we can obtain the best $P_{D,N}$ for a given $P_{F,N}$ by using a search procedure on the variables, $P_{F,i}$, $i = 1, \ldots, (N - 1)$. We have recursively used the one-dimensional optimization routine FMIN [15] for this purpose. The algorithm also requires the zero of a function in order to obtain the thresholds at each stage (7). The ZEROIN routine in [15] is used to solve for the zeros. The convergence to the optimum value is obtained in the case of 2 sensors and 3 sensors. For performance comparison, we also considered the following parallel fusion schemes: two sensors, identical thresholds at the sensors, AND, OR rules, and three sensors, identical thresholds at the sensors, AND, OR, majority logic rules. In the three-sensor case we also consider two other rules, termed $F1$ and $F2$. $F1$ corresponds to the Boolean function $u_0 = u_1 + u_2u_3$ and $F2$ corresponds to $u_0 = u_1(u_2 + u_3)$. For $F1$ and $F2$, sensors numbered 2 and 3 operate at the same thresholds. In all the cases the observations at the sensors are taken to be IID. Two channel models, namely the constant signal detection in additive white Gaussian noise (AWGN) and the detection of a slowly fluctuating Rayleigh target [3, 12] are considered.

Figs. 4–6 show the performance of two sensors in AWGN channel and Figs. 7–9 show the performance

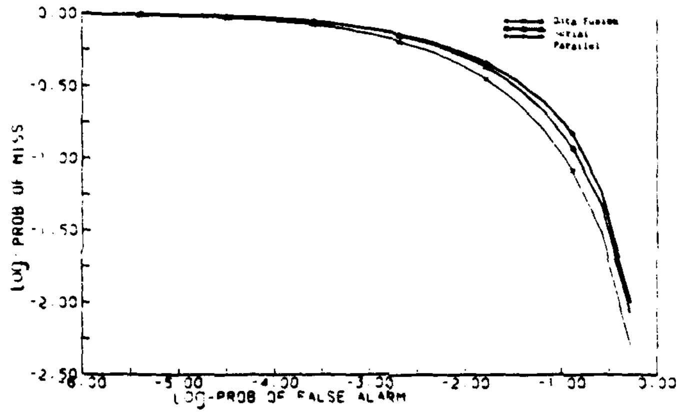


Fig. 4. Performance of serial scheme with two sensors: constant signal in Gaussian noise and signal-to-noise ratio of 5 dB

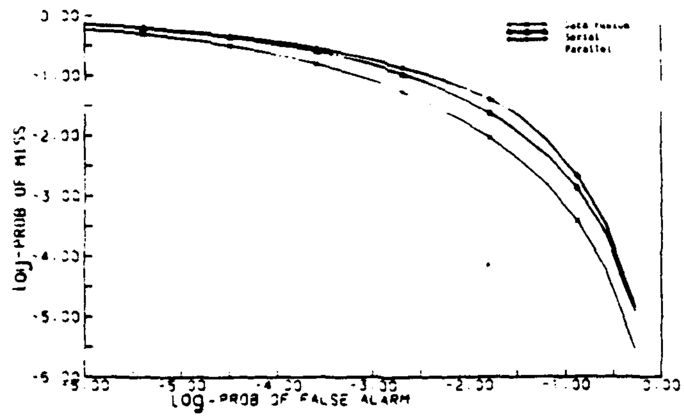


Fig. 5. Performance of serial scheme with two sensors: constant signal in Gaussian noise and signal-to-noise ratio of 10 dB

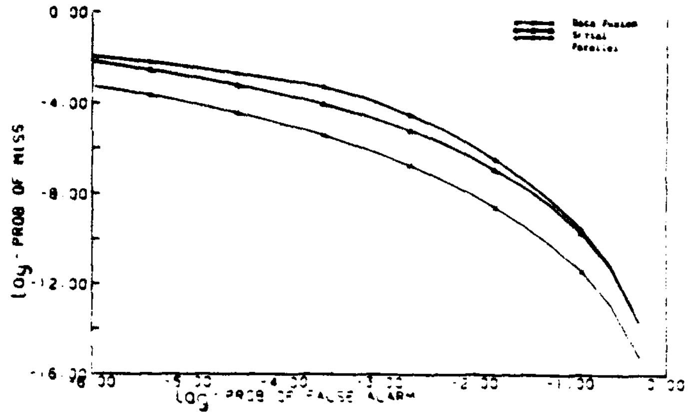


Fig. 6. **Performance of serial scheme with two sensors:** constant signal in Gaussian noise and signal-to-noise ratio of 15 dB

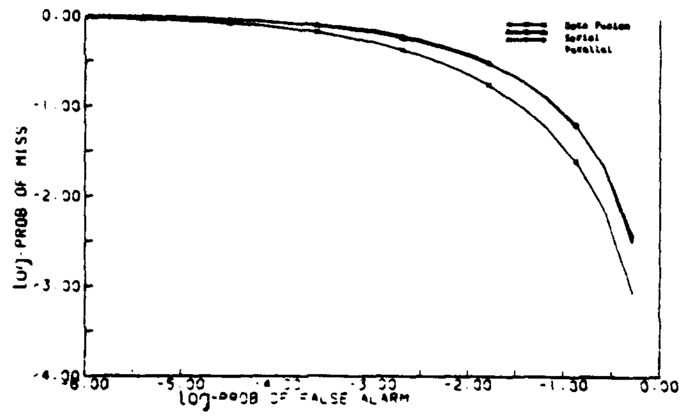


Fig. 7. **Performance of serial scheme with three sensors:** constant signal in Gaussian noise and signal-to-noise ratio of 5 dB

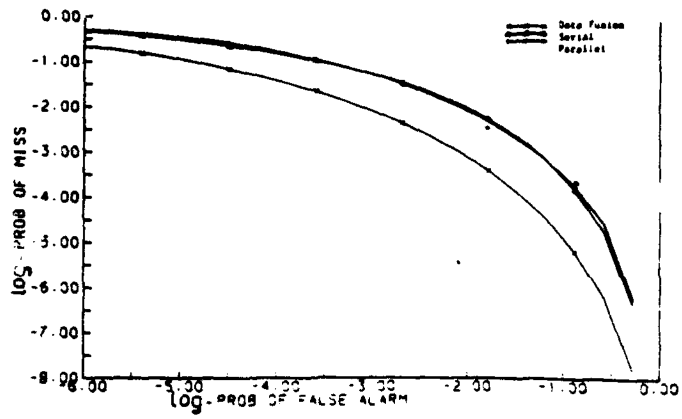


Fig. 8. **Performance of serial scheme with three sensors:** constant signal in Gaussian noise and signal-to-noise ratio of 10 dB

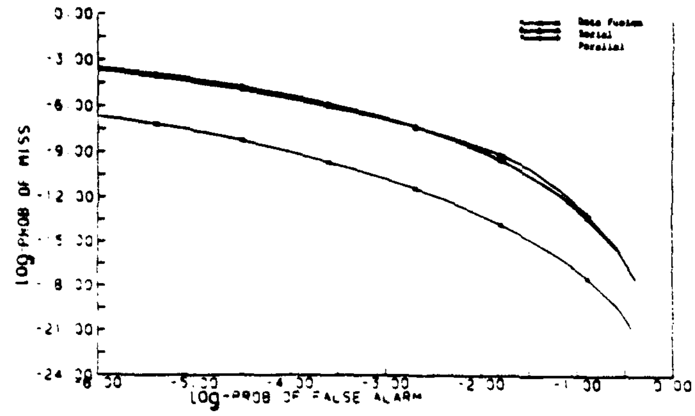


Fig. 9. Performance of serial scheme with three sensors: constant signal in Gaussian noise and signal-to-noise ratio of 15 dB

with three sensors. The curve named parallel is the best of the several parallel decision rules mentioned above and the data fusion corresponds to the centralized detection scheme which uses data available at all the sensors. With two sensors, the serial performs better than the parallel, especially at larger signal-to-noise ratios. With three sensors, the performance of the two schemes are nearly the same. Also, either of them is poor compared with the data fusion. This is due to the loss associated with the distributed detection. In Rayleigh target detection with two or three sensors, the OR rule is better than the rest of the parallel fusion rules. Moreover, the numerical computation shows that the serial is equivalent to OR for this channel. Theoretically establishing the equivalence has not been possible. In the sense that the serial is only as good as the OR rule, one can term the Rayleigh channel as conservative (Theorem 2 in the next subsection implies that the serial should be at least as good as the OR rule). Figs. 10–15 show the performances of different schemes for the Rayleigh target detection. In Figs. 13–15, the performances of $F1$ and $F2$ are equivalent and hence the corresponding graphs coincide with each other.

B. Comparison with Parallel Scheme

An optimal parallel fusion is the parallel scheme of Fig. 1 which gives the largest possible probability of detection for a given probability of false alarm at the fusion. Only a monotone increasing switching function, called the positive unate function [17], qualifies as a candidate for the optimal fusion switching function. This can be easily proved from the requirement that the optimal scheme employs likelihood ratio test at the fusion. One property of monotone increasing function is that function, when expressed as a sum of products does not contain any complemented variables. A switching function which can be expressed as a sequence of two input and one output functions is a positive unate function and hence qualifies as a candidate for the optimal parallel

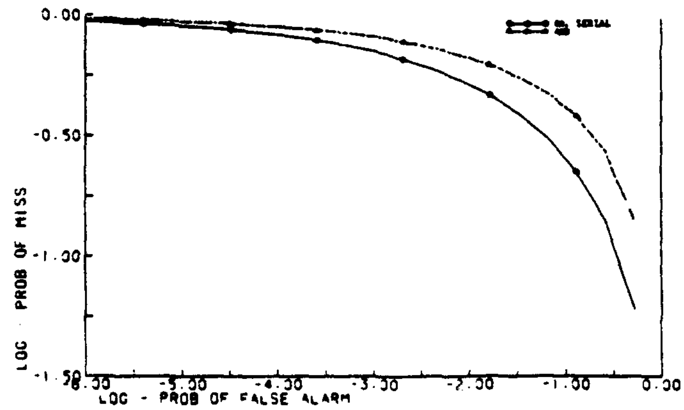


Fig. 10. Performance of serial and parallel schemes for Rayleigh target detection with two sensors: energy-to-noise density ratio of [illegible] dB

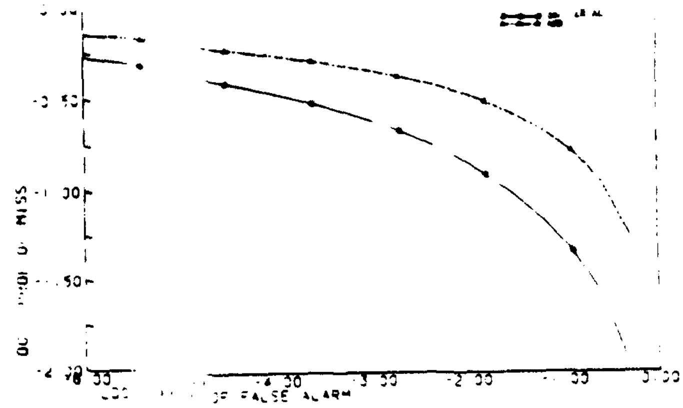


Fig. 11. Performance of serial and parallel schemes for Rayleigh target detection with two sensors: energy-to-noise density ratio of 10 dB

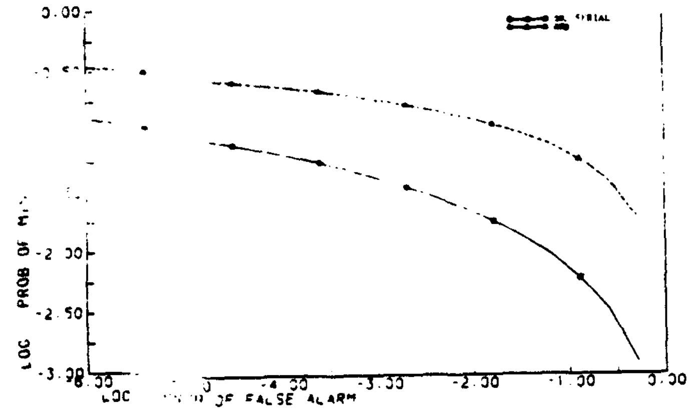


Fig. 12. Performance of serial and parallel schemes for Rayleigh target detection with two sensors: energy-to-noise density ratio of 15 dB

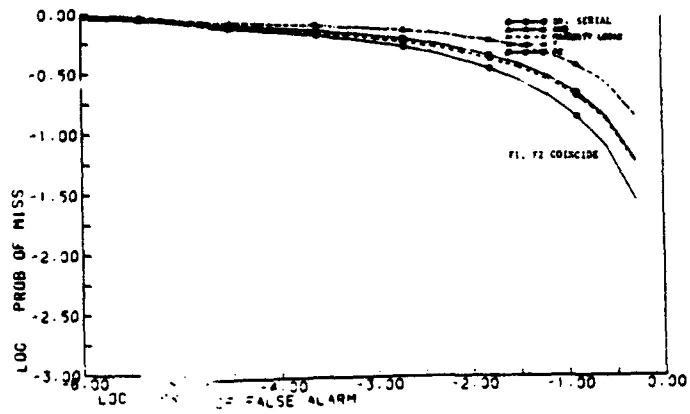


Fig. 13. Performance of serial and parallel schemes for Rayleigh target detection with three sensors: energy-to-noise density ratios of 5 dB

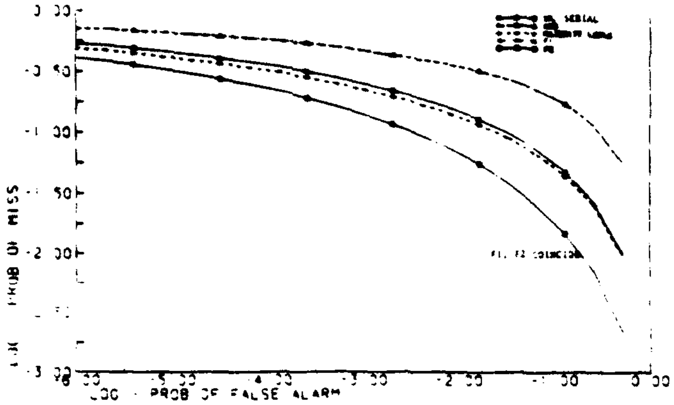


Fig. 14. Performance of serial and parallel schemes for Rayleigh target detection with three sensors: energy-to-noise density ratio of 10 dB

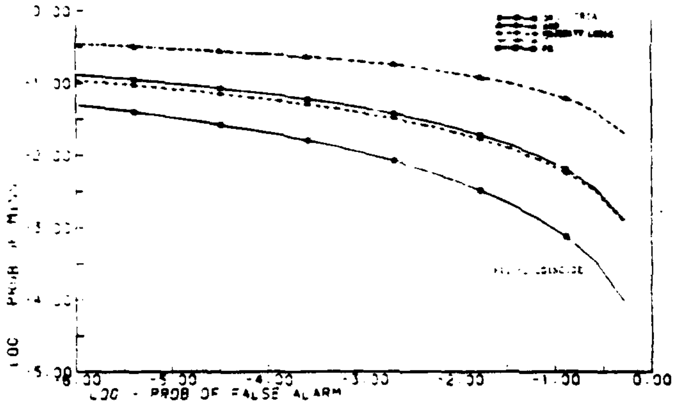


Fig. 15. Performance of serial and parallel schemes for Rayleigh target detecting with three sensors: energy-to-noise density ratio of 15 dB

fusion function. An example of one such switching function of three variables is shown in Fig. 16. Fig. 16 also shows the serial scheme with three sensors.

Theorem 2 (given below) establishes a sufficient condition for the performance of the optimal serial scheme to be not inferior to the performance of the optimal parallel scheme.

THEOREM 2. *If the switching function corresponding to the optimal parallel fusion can be realized in terms of a sequence of two variable functions with single output, then the optimal serial scheme is b‵tter than or equal to the optimal parallel scheme.*

PROOF. Consider the conservative situation in which the decision variable u_1 in Fig. 16(a) and (b) are identical and each stage of the serial scheme operates at the corresponding false alarms of the parallel scheme (in the Appendix we show that it is possible to achieve such an operation). The u_2 in Fig. 16(b) is a function of u_1 and the observation Z_2. Since the mapping of (u_1, u_2) to $\hat{u}_2$ in the parallel is contained in the mapping of (u_1, Z_2) to $\hat{u}_2$ in the serial, the detection power $P^{*}_{D,2}$ attained at $P_{F,2}$ in the serial is greater than or equal to $P_{D,2}$. Similarly, u_0 in the parallel is a function of u_2 and u_3 only whereas in the serial it is a function of $\hat{u}_2$ and the observation Z_3. It is observed that the $\hat{u}_2$ of the serial has the same false alarm $P_{F,2}$ of the parallel but has a greater than or equal power. For the serial case, the proof of Theorem 1 shows that the detection probability of any stage operating at certain false alarm is a monotone nondecreasing function of the detection probability of the previous stage operating at some false alarm. It then follows that $P^{*}_{D,0}$ s greater than or equal to $P_{D,0}$. By induction the proof is complete for any N. Conservatively it is assumed that the false alarm at each stage of the serial is identical to the one in the parallel scheme. If the serial scheme false alarms are optimized then definitely $P^{*}_{D,0}$ cannot be less than $P_{D,0}$.

From Theorem 2, we observe that for the case of two sensors, the optimal serial is better than or equal to the optimal parallel scheme. With three sensors, it is better

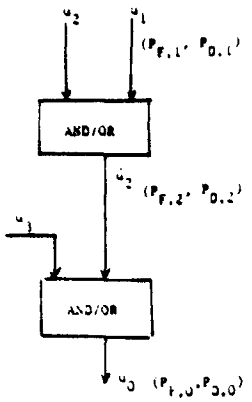


Fig. 16(a). Example of two input and one output parallel fusion function with three sensors.

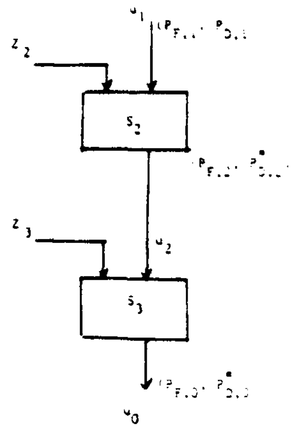


Fig. 16(b). Serial scheme with three sensors

than or equal unless the optimal parallel is a majority decision logic. In such a case, only an actual performance assessment determines which is better. As mentioned earlier, in the case of Rayleigh channel with two or three sensors, the numerical results show that the optimal serial is just equivalent to OR. In this sense the Rayleigh channel can be termed conservative. Also, in Figs. 7–9, over the range of false alarms where the parallel outperforms the serial, the best of the parallel is the majority decision rule. In the range where serial is better, the best of the parallel belongs to the class of Theorem 2.

V. CONCLUSION

A serial distributed network of N sensors detecting the presence or absence of a signal is analyzed. When the sensor observations conditioned on the hypothesis are statistically independent, the sensors employ N–P test for maximizing the detection probability for a given false alarm probability at the Nth stage (Theorem 1). For certain noise distributions, the parallel structure requiring its fusion scheme to belong to a certain class of switching functions, is not superior to the serial scheme (Theorem 2). As a drawback, any serial network is vulnerable to link failures. Some numerical examples illustrate the performance of the optimal serial decision scheme.

In the case of Rayleigh target detection with two and three sensors, the performances of the serial and the OR fusion rule are equal. For AWGN channel and two sensors, the serial performs better than the parallel. However, with three sensors the performance is essentially the same. It is not known whether there exists any channel, practical or hypothetical, such that the serial is better than the parallel for a distributed network with three or more sensors. Considering the complexity of the serial scheme and the results from this limited study, the choice seems to favor the parallel fusion for the distributed detection problem.

APPENDIX

It is shown here that any false alarm is realizable at any stage of a serial configuration. Let us denote for simplicity $P_{F,j-1}$, $P_{F,j}$, $P_{D,j-1}$, $t_{j,1}$, $t_{j,0}$, a_j, and b by α, α_0, β, t_1, t_0, a, and b, respectively. Therefore, using (2) and (3), and (7)

$$\alpha_0 = (1-\alpha)a + \alpha b$$

$$t_0 = t\,\frac{1-\alpha}{1-\beta}$$

$$t_1 = t\,\frac{\alpha}{\beta}. \tag{A1}$$

The likelihood ratio Λ (from (3)) and hence a and b are continuous functions of t. Hence, for a fixed α, α_0 is a continuous function of t. Let the support of the distribution of Λ be between t_l and t_h ($t_l \geq 0$ and $t_h \leq \infty$). As t_0 approaches t_l, a, b, and α_0 approach 1 and as t_l approaches t_h, a, b, and α_0 approach 0. Therefore, any α_0 in (0, 1) can be obtained.

Please note that the method employed here is suggested by one of the reviewers.

REFERENCES

[1] Tenney, R.R., and Sandell, N.R., Jr. (1981) Detection with distributed sensors. *IEEE Transactions on Aerospace and Electronic Systems*, AES-17 (July 1981), 501–510.

[2] Chair, Z., and Varshney, P.K. (1986) Optimal data fusion in multiple sensor detection systems. *IEEE Transactions on Aerospace and Electronic Systems*, AES-22, 1 (Jan. 1986), 98–101.

[3] Srinivasan, R. (1986) Distributed radar detection theory. *IEE Proceedings*, 133, Pt. F, 1 (Feb. 1986), 55–60.

[4] Ekchian, L.K., and Tenney, R.R. (1982) Detection networks. In *Proceedings of the 21st IEEE Conference on Decision and Control*, Dec. 1982, pp. 686–690.

[5] Viswanathan, R., Thomopoulos, S.C.A., and Tumuluri, R. (1987)

Serial decision in multiple sensor fusion.
In Proceedings of the 1987 Conference on Information Sciences and Systems Conference. The Johns Hopkins University, 1987, p. 124.

[6] Thomopoulos, S.C.A., Viswanathan, R., and Bougoulias, D.K. (1987)
Optimal decision fusion in multiple sensor system.
IEEE Transactions on Aerospace and Electronic Systems, AES-23, 5 (Sept. 1987), 644–653.

[7] Sadjadi, F.A. (1986)
Hypotheses testing in a distributed environment.
IEEE Transactions on Aerospace and Electronic Systems, AES-22, 2 (Mar. 1986), 134–137.

[8] Thomopoulos, S.C.A., Viswanathan, R., and Bougoulias, D.K.
Optimal and suboptimal distributed decision fusion.
To be published.

[9] Reibman, A.R., and Nolte, L.W. (1987)
Optimal detection performance of distributed sensor systems.
IEEE Transactions on Aerospace and Electronic Systems, AES-23, 1 (Jan. 1987), 24–30.

[10] Helstrom, C.W. (1965)
The performance of sensors connected in parallel and in coincidence.
IEEE Transactions on Communications, COM13 (1965), 191–195.

[11] VanTrees, H.L. (1968)
Detection, Estimation and Modulation Theory, Vol. I.
New York: Wiley, 1968.

[12] DiFranco, J.V., and Rubin, W.L. (1968)
Radar Detection.
Englewood Cliffs, N.J.: Prentice-Hall, 1968.

[13] Srinath, M.D., and Rajasekaran, P.K. (1979)
An Introduction to Statistical Theory of Signal Processing.
New York: Wiley, 1979.

[14] Mood, A.M., Graybill, F.A., and Boes, D.C. (1974)
Introduction to the Theory of Statistics.
New York: McGraw-Hill, 1974.

[15] Forsythe, G.E., and Malcom, M.A. (1977)
Computer Methods for Mathematical Computations.
Englewood Cliffs, N.J.: Prentice-Hall, 1977.

[16] Tsitsiklis, J., and Athans, M. (1985)
On the complexity of distributed decision problems.
IEEE Transactions on Automatic Control, AC-30, 5 (May 1985), 440–446.

[17] Harrison, M.A. (1965)
Introduction to Switching and Automata Theory.
New York: McGraw-Hill, 1965.

Ramanarayanan Viswanathan (S'81—M'83) received the B.E. (Hons.) degree in electronics and communication engineering from the University of Madras, India, in 1975, the M.E. degree with distinction in electrical communication engineering from the Indian Institute of Science, Bangalore, in 1977, and the Ph.D. degree in electrical engineering from Southern Methodist University, Dallas, Tex. in 1983.

From 1977 until 1980 he worked as a Deputy Engineer in the Digital Communication Department of the Research and Development Division of Bharat Electronics Limited, Bangalore, India. Since 1983 he has been an Assistant Professor in the Electrical Engineering Department at Southern Illinois University, Carbondale, Ill. His research interests include detection and estimation theory, statistical theory of communication and spread spectrum systems.

Stelios C.A. Thomopoulos (S'80—M'83) was born in Athens, Greece, on June 15, 1955. He received his Diploma of Electrical & Mechanical Engineering from the National Technical University of Athens in 1978, and both his M.S. and Ph.D. in electrical engineering from the State University of New York at Buffalo in 1981 and 1983, respectively.

Since 1983, he has been an Assistant Professor in the Department of Electrical Engineering at Southern Illinois University, Carbondale, Ill. His areas of interest include data networks, distributed processing, robotics, artificial intelligence, and neural networks.

Dr. Thomopoulos is a member and Licensed Engineer of the Technical Chamber of Greece.

Ramakrishna J. Tumuluri received his B.E. in electrical engineering from Gujarat University, India, in 1984 and his M.S. degree in electrical engineering from Southern Illinois University, Carbondale, Ill. in 1987. He is currently enrolled in the Ph D program in electrical engineering at Purdue University, West Lafayette, Ind.

His research interests include detection and estimation theory, computer networking, and artificial intelligence.

He is a member of Phi-Kappa-Phi and is a gold medal winner of Gujarat University, India.

Distributed Detection with Consulting Sensors and Communication Cost

Stelios C. A. Thomopoulos and Nickens N. Okello

Abstract—The problem of distributed detection with consulting sensors in the presence of communication cost associated with any exchange of information (consultation) between sensors is considered. We

Manuscript received October 26, 1990; revised May 24, 1991. This work was supported by the SDIO/IST managed by the Office of Naval Research under Contract N00014-k-0515.

The authors are with the Decision and Control Systems Laboratory, the Department of Electrical and Computer Engineering, The Pennsylvania State University, University Park, PA 16802.

IEEE Log Number 9201321.

0018-9286/92$03.00

consider a system of two sensors, $S1$ and $S2$, in which $S1$ is the primary sensor responsible for the final decision u_0, while $S2$ is a consulting sensor capable of relaying its decision u_2 to $S1$ when requested by $S1$. In the scenario that is considered, the final decision u_0 is based either on the raw data available to $S1$ only, or it may, under certain request conditions, also take into account the decision u_2 of sensor $S2$. Random and nonrandom request schemes are analyzed and numerical results are presented and compared for Gaussian and slow-fading Rayleigh channels. For each decision making scheme, an associated optimization problem is formulated whose solution is shown to satisfy certain *a priori* set design criteria that we consider essential for sensor fusion.

I. Introduction

Considerable research has been focused lately on the problem of distributed decision fusion [1]–[6] where a number of distributed sensors receive data from a common volume, come up with a first-stage decision, and then transmit their decisions to a fusion center which arrives at the final decision by fusing the sensor decisions (or some form of compact information received from the sensors). The main assumption in the bulk of the related literature is that the transmission of information from sensor to fusion (and possibly the opposite way) is done at no cost. This implies that exchange of information between the sensors and the fusion is possible at rates limited only by the physical bounds of the channel capacity. The main emphasis is then placed on determining the optimal sensor configuration (parallel, serial, or combination) [5]–[6], and the fusion logic (AND, OR, etc.) for an array of sensors [5]–[6].

The problem of team decision with risk is common in C^3 (command, control, and communications) applications [7], but not limited to those [13]. Practical application areas for team decision with risk extend to other fields, such as medical diagnosis, cryptography, etc., where exchange of information among decision-makers is not free and communication cost is a factor. The communication cost can translate into the risk of revealing one's position in C^3 applications, actual bandwidth limitations for transmission in bps (bits per second), cost in dollars of a leased communication line in commercial applications, or a consultation fee for the procurement of an expert opinion by a consultant.

The problem of distributed detection in the presence of communication cost has also been considered by Papastavrou and Athans [7]. In their formulation, they considered symmetric operation schemes for both the primary and the consulting sensors, in a way that ignorance could be the end result of an exchange of information between the sensors even if a price tag was associated with the information exchange. A general cost was then attached to each decision under the tested hypotheses, and the likelihood-test was shown to be the optimal decision rule under the given operating schemes [8].

In this note, we consider the problem of distributed decision making with two consulting sensors in which every inter-sensor communication incurs some risk, thus making continuous sensor communication a very expensive and prohibitive proposition. We are interested in determining the optimum decision scheme when the structure of the consultation scheme is specified given that a certain amount of risk (or communication cost) can be tolerated. Given the structure of the consultation scheme, we seek optimal decision rules that minimize cost functionals that involve the probability of false alarm, the communication cost, and the probability of miss. Different possible formulations are being discussed in this note.

II. Team Decision Schemes

The team-decision scenarios that we analyze in this note consist of a dual-sensor system and binary hypothesis testing as

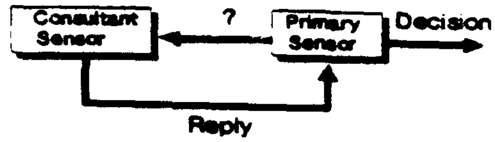


Fig. 1. Dual-sensor configuration in consultation.

in Fig. 1. Due to bandwidth limitations and the sensitivity of the data, no transmission of raw data between the two sensors is allowed. The sensors only exchange request signals and decisions. (Additional quality information bits, such as the degree of confidence associated with each decision, could have also been included in the scenarios that are considered without affecting the structure of the tests significantly.) We present numerical and some analytical results only for the cases where the primary sensor transmits request signals to the consultant sensor, whereas the latter relays only its binary decisions back to the primary, and no exchange of quality information bits takes place. Random consultation and nonrandom consultation schemes are considered.

In the analysis that follows, we assume that the probability distributions of the observations for both sensors under either hypotheses are *absolutely continuous* with respect to the Lebesgue measure and that the associated likelihood ratios are *piecewise continuous* functions of the thresholds. Furthermore, we assume that the decisions of the primary and consulting sensors are *mutually independent conditioned on each hypothesis*. Numerical evaluation of the optimal solutions for different formulations is performed in additive Gaussian noise channels [9] and slow-fading Rayleigh channels [3], [10]. The following notations will be used in the sequel.

Notations

- P_{D_i} := Detection probability of sensor Si operating alone, $i = 1, 2$.
- P_{M_i} := Miss probability of sensor Si operating alone, $i = 1, 2$.
- P_{F_i} := False alarm probability of sensor Si operating alone, $i = 1, 2$.
- P_{D12} := Detection probability of $S1$ and $S2$ in consultation.
- P_{M12} := Miss probability of $S1$ and $S2$ in consultation.
- P_{F12} := False alarm probability of $S1$ and $S2$ in consultation.
- P_D := Team detection probability.
- P_M := Team miss probability.
- P_F := Team false alarm probability.
- P_R := Request probability (it determines the consultation level).

In the nonrandom consultation case, explicit reference to the sensor threshold(s) will be required. The notation $P_X(t_i') = P_{Xi}'$ and $P_X(t_i'') := P_{Xi}''$, $X := F, M$, or D and $i = 1, 2$, will be used to indicate the false alarm ($X = F$), miss ($X = M$), or detection ($X = D$) probabilities of sensor Si operating at thresholds t_i' or t_i''. The notation P_{Xi}' and P_{Xi}'' will be used to make the expressions more compact when needed.

III. Random Consultation Schemes

A. Random Consultation with Fixed Probability and Reprocessing: Problem Formulation

The primary sensor $S1$ consults $S2$ randomly with a fixed probability of request P_R. When $S2$ is consulted, it relays its decision to $S1$, which in turn reprocesses it with its own raw data in order to come up with the final decision. The objective is to minimize the team miss probability P_M (equivalently, maximize the probability of detection P_D) for fixed false alarm probability P_F. The distinguishing feature of this scheme is that the decision to consult is random and is made independently of the degree of confidence that sensor $S1$ may have on its initial decision u_1. The major advantages of the scheme are that: a) it is simple to analyze; and b) its performance *does not depend on the prior probabilities* of the two hypotheses which may very often be unknown in C^3 and other applications.

The optimal random consultation scheme is equivalent to switching between the ROC (receiver operating characteristic) curve of $S1$ alone [9] and the ROC of the serial combination of $S1$ and $S2$ [6] (Fig. 2) according to a specified request probability P_R, so that the probability of detection is maximized for a fixed team false alarm probability α_0. (For the reader's convenience, the optimal decision test for serially connected sensors is summarized in the Appendix.) The team probabilities are easily obtained as

$$P_D = P_{D1}(1 - P_R) + P_{D12}P_R \tag{1}$$

$$P_M = P_{M1}(1 - P_R) + P_{M12}P_R \tag{2}$$

$$P_F = P_{F1}(1 - P_R) + P_{F12}P_R \tag{3}$$

where 1 in the subscript indicates the sensor $S1$ operating alone, and 12 the serial combination of $S1$ and $S2$ to be designated as $S12$ hereafter. The random consultation decision problem is mathematically formulated as follows:

$$\text{Maximize } P_D \text{ s.t. } P_F = \alpha_0 \text{ and } 0 \le P_R \le \beta_0. \tag{P1}$$

Using Lagrange multipliers ω_1 and ω_2, the constrained maximization problem ($F1$) is converted into the unconstrained maximization problem

$$\max J = P_D + \omega_1[\alpha_0 - P_F] + \frac{1}{2}\omega_2[(\beta_0 - P_R)P_R - \mu^2] \tag{P1.1}$$

where μ^2 is a positive slack variable that is used to convert the inequality constraints on P_R into an equivalent equality constraint. The maximization in (P1.1) is understood with respect to the choice of operating points of $S1$ and $S2$, and the level of consultation P_R.

B. Random Consultation Optimal Solution

Theorem 1: If the ROC's of $S1$, $S2$, and the serial combination of $S1$ and $S2$, $S12$ [6] are strictly concave, then the optimal solution to problem (P1.1) and thus (P1) involves a Neyman–Pearson (N–P) test under either stand-alone or serial modes of operation. The optimal operating points are given as solutions to the equations

$$\frac{\partial P_{D1}}{\partial P_{F1}} = \frac{\partial P_{D12}}{\partial P_{F12}} = \omega_1 \tag{4}$$

$$\alpha_0 = P_{F1}(1 - P_R) + P_{F12}P_R \tag{5}$$

$$P_R = \frac{(P_{D12} - P_{D1}) + \omega_1(P_{F1} - P_{F12})}{\omega_2} + \frac{\beta_0}{2} \tag{6}$$

$$\omega_2\mu = 0. \tag{7}$$

Hence, the optimal solution involves two N–P tests operating at points of the $S1$ ROC and the $S12$ ROC with equal slopes that satisfy $P_F = \alpha_0$ and $0 \le P_R \le \beta_0$. Condition (7) along with (6) implies that $P_R = \beta_0$ when $\omega_2 \neq 0$, which is true if $P_{D12} > P_D$. The solution $\omega_2 = 0$ implies $P_R = 0$, which is the solution when $P_{D12} = P_{D1}$ and $P_{F12} = P_{F1}$. Furthermore, under the continuity assumptions, the optimal solution is unique.

Proof: Under the assumption that the ROC's of $S1$, $S2$, and the serial combination $S12$ are strictly concave, the N–P test maximizes the probability of detection at each one for any fixed false alarm probability [9]. Thus, for any P_{F1} and P_{F12} that satisfy the constraint $P_F = \alpha_0$, and for any P_R, the detection probability is maximized if the N–P test is used under both the stand-alone and serial modes of operation. Substituting P_D and P_F in (P1.1) from (1) and (3), and differentiating J with respect to P_{F1} and P_{F12}, (4) is obtained. Differentiating J with respect to P_R, setting the result equal to zero and solving for P_R, (6) is obtained. Differentiation of (P1.1) with respect to μ results in (7).

From (7), it follows that $\mu^2 = (P_R - \beta_0)P_R = 0$ when $\omega_2 = 0$. However, from (6), $\omega_2 \neq 0$ implies that $P_R \neq 0$. Hence, $P_R = \beta_0$ in order to satisfy $\mu^2 = 0$. On the other hand, from (6), $\omega_2 = 0$ if and only if $P_{D12} = P_{D1}$ and $P_{F12} = P_{F1}$, in which case $P_R = 0$, and thus $\mu = 0$ as well.

The uniqueness of the optimal solution follows from the absolute continuity assumption and the concavity of the ROC, from which it follows that $(\partial P_{D1}/\partial P_{F1})$ and $(\partial P_{D12}/\partial P_{F12})$ are strictly monotonic functions. Hence, for each ω_1, there exist unique points on $S1$ ROC and on $S12$ ROC for which (4)–(6) are satisfied. $\square$

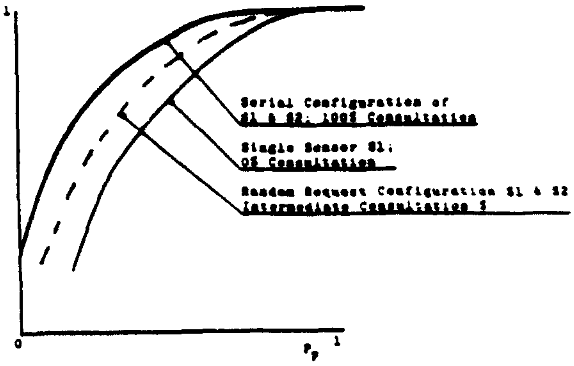


Fig. 2. Receiver operating characteristics (ROC) for different levels of random request.

C. Numerical Results

Numerical results of the optimal solution to problem (P1) in additive Gaussian noise channels and slow-fading Rayleigh channels are given in Fig. 3 for different request rates. The numerical results throughout the note are obtained assuming the following statistical models for the two channels.

Gaussian:

Observation model at each sensor: $r \sim G(0,1)$: H_0, and $r \sim G(s,1)$: H_1, where $G(\alpha, \beta)$ designates an α mean and variance β Gaussian distribution. If t_b is the threshold at the sensor, the operating false alarm and detection probabilities (P_F, P_D) are given by

$$\textit{False alarm probability}: P_F = Q(t_b)$$

$$\textit{Detection probability}: P_D = Q(t_b - \sqrt{\epsilon}) = Q[Q^{-1}(P_F) - \sqrt{\epsilon}]$$

where $Q() = 1 - \Phi()$ is the cumulative distribution function

(CDF) of the standard normal, Q^{-1} its inverse, and ϵ = SNR at the sensor in decibels.

Rayleigh:

$$\textit{False alarm probability: } P_F = [\lambda(1+\epsilon)]^{-(1+\frac{1}{\epsilon})}$$

$$\textit{Detection probability: } P_D = [P_F]^{(\frac{1}{1+\epsilon})}$$

where λ is the threshold used, and ϵ the SNR at the sensor in decibels.

From Fig. 3, it is easy to see that the optimal solution to problem (P1) is: a) monotonic with respect to the information fused independent of the quality of the sensors; b) monotonic with respect to β_0; and c) independent of the *a priori* uncertainty. These properties are analytically proven in [12].

D. Random Consultation Suboptimal Solution

A suboptimal solution to problem (P1) is obtained if P_{F1} and P_{F12} are constrained to be equal, thus equal to α_0 according to (3). The suboptimal solution to problem (P1) involves N-P tests for both $S1$ and $S12$ as well. The suboptimal operating point is given as a point between the $S1$ and $S12$ ROC curves at level α_0 determined by the equality $P_R = \beta_0$ (Fig. 2). The system $P_D = P_{D1}(1-\beta_0) + P_{D12}\beta_0$ [12]. Numerical results of the suboptimal solution to (P1) in Gaussian and slow-fading Rayleigh channels are shown in Fig. 4 for $\beta_0 = 0.25$ and 0.75. For comparison, the optimal random consultation ROC's for the same values of β_0 are overlayed in the same figure. The ROC's of the optimal random consultation scheme are slightly (but visibly) superior to the ROC's of the suboptimal scheme for the Rayleigh channel, but almost identical (superior only on the third significant digit, not visible in the plots) to the suboptimal scheme for the Gaussian channel.

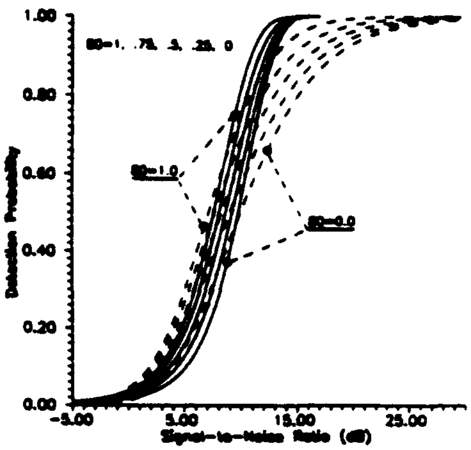


Fig. 3. Optimal random consultation detection probability versus SNR for false alarm probability $A0 = 0.001$ and different request rates. Channels: Gaussian (solid) and slow-fading Rayleigh (dashed).

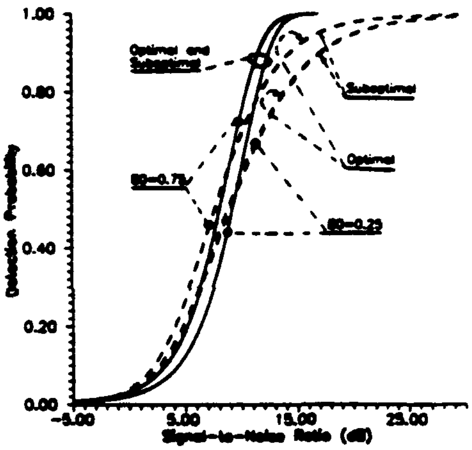


Fig. 4. Comparison of detection probabilities for the suboptimal nonrandom scheme for request probabilities 0.25 and 0.75. Channels: Gaussian (solid) and Rayleigh (dashed).

IV. Nonrandom Consultation Schemes Without Reprocessing

A. Operating Scenario

In the nonrandom consultation schemes we assume that the decision to consult is made only when the initial decision u_1 of $S1$ falls within the indecision region (see below for definition), otherwise, u_1 is taken as final if it falls outside the region of indecision. While several different operating scenarios are possible, we are only concerned with the case in which $S1$ may consult $S2$ but does not relay any quality information regarding its initial findings. When requested, $S2$ processes its own raw data taking into account the fact that it has been consulted, and transmits its decision u_2 to $S1$ which then treats it as the final decision. Hence, no reprocessing takes place at $S1$ after consultation.[1]

We constrain the consultation schemes to the following class. Let $\Lambda_i(r_i) := (P(r_i|H_1)/P(r_i|H_0))$ designate the likelihood ratio (LR) at the ith sensor using data r_i, $i = 1, 2$. Assume that $S1$ has an uncertainty region (t_1', t_1''). When $\Lambda_1(r_1) > t_1''$, $S1$ decides in favor of H_1. When $\Lambda_1(r_1) < t_1'$, $S1$ decides in favor of H_0. In either case, no consultation takes place and the decision of $S1$ is final. When $\Lambda_1(r_1) \in (t_1', t_1'')$, $S1$ consults $S2$ without transmitting any quality information about its preliminary decision to $S2$. When $S2$ is consulted, it processes its data using an LRT conditioned on the event that $S1$'s decision falls in the indecision region I, induced by the fact that it has been consulted, and relays its decision u_2 to $S1$ which takes it as final for the entire system. Thus, $S1$ decides according to the following scheme:

$$\Lambda_1(r_1) \le t_1': \text{choose } H_0$$
$$t_1' < \Lambda_1(r_1) < t_1'': \text{choose } I \text{ (Ignorance)}$$
$$\Lambda_1(r_1) \ge t_1'': \text{choose } H_1 \tag{8}$$

while $S2$ employs the familiar likelihood ratio test given by

$$\Lambda_2(r_2, u_1 = I) \underset{H_0}{\overset{H_1}{\gtrless}} t_2. \tag{9}$$

If u_0 denotes the final decision of the system, the overall miss

[1] A more symmetric scenario than the one used in random consultation would call for reprocessing of u_2 by $S1$ along with its own raw data during consultation. However, the performance of the symmetric scenario would be very close to the nonsymmetric, nonrandom request scheme considered in this note, as it can be seen from Fig. 9 where the performance of the serial scheme (which corresponds to the optimal, nonrandom request scheme when the consultation rate is 100%) is compared to the optimal, nonsymmetric, nonrandom request scheme at optimal consultation rate.

probability P_M is given by

$$\begin{aligned} P_M &= P(u_0 = 0|H_1) \\ &= \sum_{u_1} P(u_0 = 0|u_1, H_1)P(u_1|H_1) \\ &= P(u_0 = 0|u_1 = 1, H_1)P(u_1 = 1|H_1) \\ &\quad + P(u_0 = 0|u_1 = 0, H_1)P(u_1 = 0|H_1) \\ &\quad + P(u_0 = 0|u_1 = I, H_1)P(u_1 = I|H_1). \end{aligned} \tag{10}$$

The first part of the right-hand side of (10) equals zero, while the second and third parts can be simplified to give

$$P_M = P(u_1 = 0|H_1) + P(u_2 = 0|u_1 = I, H_1)P(u_1 = I|H_1). \tag{11}$$

Expressing P_M in terms of the likelihood ratio $\Lambda_1(r_1)$ and $\Lambda_2(r_2, u_1 = I)$, we get

$$\begin{aligned} P_M &= \int_{\Lambda_1 < t_1'} dP(\Lambda_1(r_1)|H_1) \\ &\quad + \int_{\Lambda_2 < t_2} dP(\Lambda_2(r_2, u_1 = I)|H_1) \int_{t_1'}^{t_1''} dP(\Lambda_1(r_1)|H_1). \end{aligned} \tag{12}$$

Dropping the arguments from the LR's $\Lambda_1(r_1)$ and $\Lambda_2(r_2, u_1 = I)$ for notational compactness, an expression for P_D is obtained from (11)

$$P_D = \int_{\Lambda_1 > t_1''} dP(\Lambda_1|H_1) + \int_{\Lambda_2 > t_2} dP(\Lambda_2|H_1) \int_{t_1'}^{t_1''} dP(\Lambda_1|H_1). \tag{13}$$

Similarly, for the overall false alarm probability we obtain

$$P_F = \int_{\Lambda > t_1''} dP(\Lambda_1|H_0) + \int_{\Lambda_2 > t_2} dP(\Lambda_2|H_0) \int_{t_1'}^{t_1''} dP(\Lambda_1|H_0) \tag{14}$$

and for the probability of request

$$P_R = \int_{t_1'}^{t_1''} dP(\Lambda_1) = \int_{t_1'}^{t_1''} [dP(\Lambda_1|H_0)P_0 + dP(\Lambda_1|H_1)(1 - P_0)]. \tag{15}$$

Note that it is necessary to express the likelihood ratio $\Lambda_2(r_2, u_1 = I)$ in terms of $\Lambda_2(r_2)$ in order to be able to evaluate the integrals $\int_{\Lambda_2 > t_2} dP(\Lambda_2|H_0)$ and $\int_{\Lambda_2 > t_2} dP(\Lambda_2|H_1)$. Taking into account the assumption that u_1 and u_2 are independent conditioned on each hypothesis

$$\begin{aligned} \Lambda_2(r_2, u_1 = I) &= \frac{P(r_2, u_1 = I|H_1)}{P(r_2, u_1 = I|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} t_2 \\ &= \frac{P(r_2|H_1)P(u_1 = I|H_1)}{P(r_2|H_0)P(u_1 = I|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} t_2 \end{aligned} \tag{16}$$

or

$$\Lambda_2(r_2, u_1 = I) = \Lambda_2(r_2) \frac{\int_{t_1'}^{t_1''} dP(\Lambda_1|H_1)}{\int_{t_1'}^{t_1''} dP(\Lambda_1|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} t_2. \tag{17}$$

Hence

$$\Lambda_2(r_2) \underset{H_0}{\overset{H_1}{\gtrless}} t_2 \frac{\int_{t_1'}^{t_1''} dP(\Lambda_1|H_0)}{\int_{t_1'}^{t_1''} dP(\Lambda_1|H_1)} = t_2'. \tag{18}$$

Therefore, it follows that

$$\int_{\Lambda_2 > t_2} dP(\Lambda_2(r_2, u_1 = I)|H_i) = \int_{\Lambda_2 > t_2'} dP(\Lambda_2(r_2)|H_i), \quad \text{for } i = 0, 1. \tag{19}$$

Using (19) and the more convenient notation with the thresholds t_1', t_1'' of $S1$, and t_2' of $S2$ explicitly indicating the correspondence between the mode of operation and the related probabilities (12)–(15) take on the more compact form

$$P_M = P_M(t_1') + P_M(t_2')[P_M(t_1'') - P_M(t_1')] \tag{20}$$

$$P_D = P_D(t_1'') + P_D(t_2')[P_D(t_1') - P_D(t_1'')] \tag{21}$$

$$P_F = P_F(t_1'') + P_F(t_2')[P_F(t_1') - P_F(t_1'')] \tag{22}$$

$$P_R = P_0[P_F(t_1') - P_F(t_1'')] + (1 - P_0)[P_M(t_1'') - P_M(t_1')]. \tag{23}$$

Note that the expressions for P_R, P_F and P_M are subject to the constraint $t_1'' \geq t_1'$ which in turn implies that

$$P_F(t_1') \geq P_F(t_1'') \text{ and } P_M(t_1') \leq P_M(t_1''). \tag{24}$$

In the nonrandom consultation framework described above, the team-decision problem can be formulated as a constrained or unconstrained optimization problem. A number of different formulations are meaningful depending on the application and the objective. Using (20)–(23), and the constraint (24), it is possible to determine the optimum thresholds t_1', t_1'', and t_2' numerically for a wide range of formulations. In this note however, we are only concerned with one nonrandom consultation formulation. Additional nonrandom consultation formulations and numerical results are available in [12].

B. Problem Formulation

We formulate the nonrandom consultation decision making problem as follows.

Maximize:

$$P_D \text{ subject to } P_F = \alpha_0 \text{ and } P_R \leq \beta_0. \tag{P2}$$

The inequality constraint in P_R and the N–P test optimal solution to each subproblem of $S1$ operating alone or $S2$ in consultation with $S1$, guarantee the existence of the optimal solution. However, the optimal solution to problem (P2) cannot be obtained analytically. Using numerical techniques, the optimal solution to problem (P2) (i.e., the optimal thresholds) can be obtained via a search algorithm. Using the more compact notation P_{Xi}' and P_{Xi}'' from the earlier defined notations, (22) and (23) are written, respectively, as

$$\alpha_0 = P_{F1}'' + P_{F2}'[P_{F1}' - P_{F1}''] \tag{25}$$

and

$$P_R = P_0[P_{F1}' - P_{F1}''] + (1 - P_0)[P_{M1}'' - P_{M1}'] \leq \beta_0. \tag{26}$$

The maximum P_D is found by searching over P_{F1}'' in the range of [0, 1] and using (25) and (26) to determine P_{F1}' and P_{F2}', subject to the constraint $P_{F1}' \geq P_{F1}''$ (since $t_1'' \geq t_1'$).

Lemma 1: Let t_{1,α_0} be the optimal threshold of $S1$ for problem (P2) when $\beta_0 = 0$, i.e., when $S1$ operates alone at false alarm probability α_0. Then, for every $\beta_0 > 0$

$$t_1' > t_{1,\alpha_0} \le t_1'' \tag{27}$$

where t_1' and t_1'' are the thresholds of $S1$ for the optimal solution to problem (P2). Furthermore, in order to improve the performance of the consultation arrangement beyond that of $S1$ operating alone at the same false alarm probability, the optimal threshold for $S2$ must satisfy the inequality

$$\int_{\Lambda_2 > t_2'} dP(\Lambda_2(r_2)|H_1) > \frac{\int_{t_{1,\alpha_0}}^{t_1''} dP(\Lambda_1|H_1)}{\int_{t_1'}^{t_1''} dP(\Lambda_1|H_1)}. \tag{28}$$

If there are no (t_1', t_1'', t_2') such that (28) is satisfied as a strict inequality, $P_R = 0$, $t_1' = t_1'' = t_{1,\alpha_0}$, and $P_D = P_{D1}(t_{1,\alpha_0})$.

Proof: From $P_F = \alpha_0$ it follows that $t_1'' \ge t_{1,\alpha_0}$. Equating the false alarm probability of the two-sensor system with that of $S1$ operating alone, it follows easily through elementary algebraic manipulations that

$$\int_{\Lambda_2 > t_2'} dP(\Lambda_2(r_2)|H_0) = \frac{\int_{t_{1,\alpha_0}}^{t_1''} dP(\Lambda_1|H_0)}{\int_{t_1'}^{t_1''} dP(\Lambda_1|H_0)} \le 1 \tag{29}$$

from which (27) follows. Using (21), the requirement $P_D > P_{D1}(t_{1,\alpha_0})$ translates to (28) with some elementary algebra. If (28) cannot be satisfied as a strict inequality, it implies that for every (t_1', t_1'') the ratio on the RHS of (28) must always be one, since the LHS of (28) is a cumulative probability distribution which by assumption is assumed to be a continuous function of the threshold t_2', thus taking all the values in [0, 1]. This in turn implies that $t_1' = t_1'' = t_{1,\alpha_0}$, from which it follows that $P_R = 0$ and, hence, $P_D = P_{D1}(t_{1,\alpha_0})$. □

C. Numerical Results

The optimization problem (P2) was solved numerically in the Gaussian and slow-fading Rayleigh channels for different maximum allowable request rates β_0. Numerical results from the two channels for fixed team false alarm probability $\alpha_0 = 10^{-3}$ are summarized in Figs. 5 and 6. The detection probability curves for the two channels were obtained by constraining the maximum allowable request probability at a designated level β_0 and numerically solving the optimization problem (P2). On each figure, the request probability envelope (bell-shaped curve) indicates the maximum optimal consultation rate and is achieved by setting $\beta_0 = 1$. It is interesting to note from Figs. 5 and 6 that the nonrandom consultation strategy does not always use the maximum allowable consultation rate for the entire SNR range. This seems to be counterintuitive, since it can be argued that more often consultation can only improve the team performance. This might have been true if the decision to consult were not associated with the degree of confidence of the primary sensor on its preliminary findings. However, in the nonrandom strategy scenario that we consider here, this is not the case. In our scenario, the decision to consult is associated with the confidence that the primary sensor has on its data. Furthermore, since the decision of the consulting sensor is taken to be final once consultation takes place, the initial decision of the primary sensor only affects the threshold of the secondary sensor (18). Thus, the maximum consultation rate is not necessarily always equal to the maximum allowable request rate, for the maximum

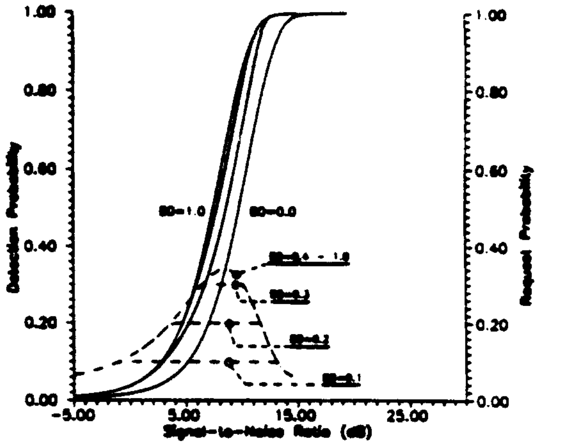


Fig. 5. Detection (solid) and optimal nonrandom request (dashed) probabilities versus SNR for a Gaussian channel. False alarm probability $A0 = 0.001$ and prior probability $P0 = 0.5$.

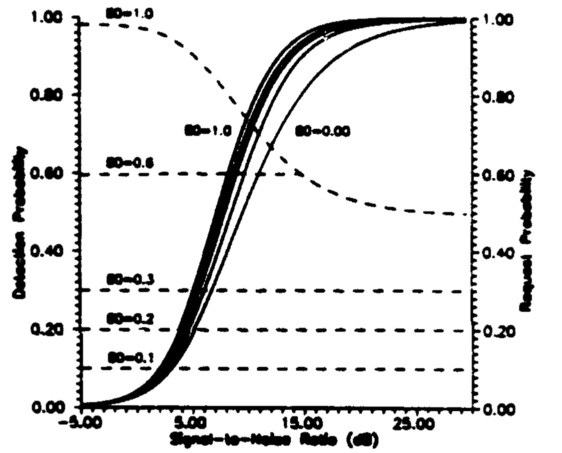


Fig. 6. Detection (solid) and optimal nonrandom request (dashed) probabilities versus SNR for the slow-fading Rayleigh channel. False alarm probability $A0 = 0.001$ and prior probability $P0 = 0.5$.

consultation rate is dictated by the degree of confidence of the primary sensor on its initial findings which is a function of the SNR and the channel statistics. From Figs. 5 and 6, it is seen that the optimal maximum request probability saturates at different levels for the two channels. This difference in the behavior of the two channels is explained in [13].

Another observed difference in the behavior of the two channels is reflected on the variation of the maximum optimal consultation rate with the prior P_0 for fixed SNR and false alarm probability, Fig. 7. The request probability P_R in (23) depends on the probability masses associated with the indecision region under each hypothesis and on the prior P_0. If the inequality constraint $P_R \le \beta_0$ can be satisfied as a strict inequality for any prior P_0, then the indecision probability masses $[P_F(t_1') - P_F(t_1'')]$ and $[P_M(t_1'') - P_M(t_1')]$ will remain constant irrespective of P_0. This is definitely the case when $\beta_0 = 1$. Hence, for the maximum optimal consultation the variation of P_R with respect to P_0 is linear (Fig. 7). For the Rayleigh channel, the maximum

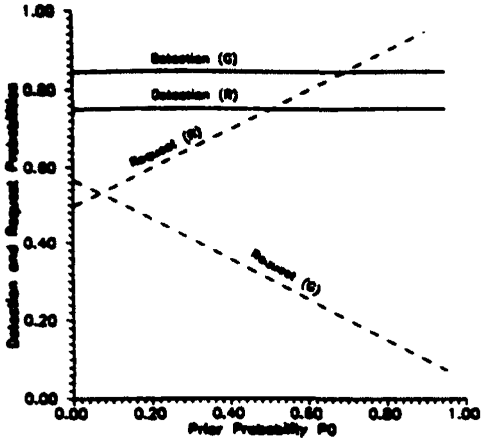


Fig. 7. Effect of prior probability $P0$ on detection and optimal request probabilities. $A0 = 0.001$, SNR = 10.0 dB. Channel: Gaussian (G) and Rayleigh (R).

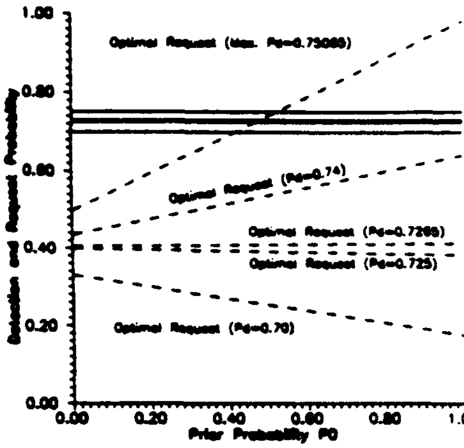


Fig. 8. Optimal request rates required to achieve specific detection probabilities for the Rayleigh channel at 10.0 dB. Slope of optimal request rate changes sign depending on the specified detection probability.

optimal request rate is monotonically increasing with P_0, while it is monotonically decreasing for the Gaussian channel. For the Gaussian channel, it was found that P_R decreases as P_0 increases irrespective of β_0 and of SNR. However, for the Rayleigh channel, the variation of P_R as P_0 increases depends on β_0 and on SNR. The reason is that the slope of the line that determines P_R in (23), that is $[P_F(t_1') - P_F(t_1'')] - [P_M(t_1'') - P_M(t_1')]$, does not maintain the same sign for all β_0's and SNRs. From Fig. 8, it is seen that the slope is negative, implying a decreasing consultation rate for $P_D \leq 0.725$ but positive, implying an increasing slope for $P_D > 0.725$.

From the analysis of the numerical results, it follows that despite the exhibited differences between the two channels, the optimal solutions possess the desired properties postulated by the design criteria in [12]. Analytical results supporting some of the above qualitative statements for channels that can be modeled by absolutely continuous distributions with respect to the Lebesgue measure under either hypothesis can be found in [12] for the formulation of problem (P2) and other formulations.

D. Comparison of Numerical Results

In order to compare the advantages from nonrandom consultation versus random consultation, the minimum necessary request probability for achieving the same detection probability with optimal nonrandom consultation as with optimal random consultation is computed and plotted in Fig. 8 as function of the prior probability for the Rayleigh channel assuming team false alarm probability 0.001. For random consultation, the request probability is independent of the prior P_0. On the other hand, the optimal request rate for nonrandom consultation increases linearly as P_0 increases for $P_D > 0.725$, but remains substantially below the required request rate in random consultation for the same P_D (compare to Fig. 3). Thus, optimal nonrandom consultation results in substantial reduction in communication requirements (consultation rate) required to achieve a certain team performance level compared to random consultation. Notice that in Fig. 7, the detection probability for the Rayleigh channel is below 0.725, and thus the request rate decreases as the prior probability increases, in agreement with the results in Fig. 8. If a cost factor (price) is associated with the communication requirements, the ($P2$) optimization problem can be modified to account for that cost [12].

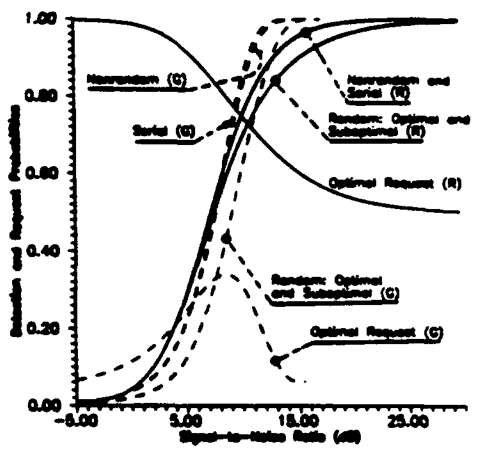


Fig. 9. Comparison of serial, optimal nonrandom, and optimal and subooptiomal random schemes at optimal request rates for Gaussian (G) and Rayleigh (R) channels. $A0 = 0.001$ and $P0 = 0.5$.

The nonrandom consultation scheme is compared to optimal and suboptimal consultation schemes for the request rates equal to the optimal nonrandom request rate, i.e., the rate that corresponds to $\beta_0 \leq 1$ (Fig. 9). The optimal symmetric, nonrandom consultation scheme for $\beta_0 = 1$, i.e., the serial combination $S12$, is also included in the figure. The following are observed: a) the optimal, nonsymmetric, nonrandom consultation scheme performs very closely (identically in the case of the Rayleigh channel) to the optimal, symmetric, nonrandom consultation scheme, i.e., the serial combination $S12$; b) the performance of the optimal and suboptimal random consultation schemes is inferior to the nonrandom consultation at the optimal request rate; and c) the suboptimal random consultation scheme performs worse than the optimal random consultation for the Gaussian channel but identically to it for the Rayleigh channel.

Conclusions

Random and nonrandom consultation schemes are examined and different mathematical formulations of the decision making problem in the presence of consultation cost are analyzed. The problem of consulting sensors is cast in a general framework suggested for sensor integration that satisfies design criteria that guarantee the benefits of data fusion [11]. The analysis and the numerical results indicate that the optimal solutions to the different schemes introduced in this note satisfy the three data fusion design criteria which we advocate to be essential for the design of any practical decision making system, namely *monotonicity* with respect to fused information, *monotonicity* with respect to the cost associated with aquiring the information, and *robustness* with respect to *a priori* uncertainty. Comparison between the random and nonrandom consultation schemes demonstrates that nonrandom consultation considerably reduces the communication requirements for achieving a desired performance level compared to the communication requirements for achieving the same performance level with random consultation. Additional analytical and numerical results from different formulations of the problem can be found in [12].

Appendix

To derive the ROC of the serial combination of $S1$ and $S2$, we consider a system of two sensors $S1$ and $S2$ in which the decision u_2 of sensor $S2$ is transmitted to sensor $S1$ and is then used together with the raw data Z_1 available to $S1$ to arrive at a final decision u_1. To that extent, we follow an analysis similar to [6]. Denoting the distribution of r_1 as $p(r_1|H_0)$ and $p(r_1|H_1)$, the likelihood ratio at sensor $S1$ becomes

$$\frac{L(r_1, u_2|H_1)}{L(r_1, u_2|H_0)} = \frac{p(r_1|H_1)[P_{D2}\,\delta(u_2 - 1) + (1 - P_{D2})\,\delta(u_2)]}{p(r_1|H_0)[P_{F2}\,\delta(u_2 - 1) + (1 - P_{F2})\,\delta(u_2)]} \tag{A.1}$$

where

$$P_{D2} = Pr(u_2 = 1|H_1) \text{ and } P_{F2} = Pr(u_2 = 1|H_0) \tag{A.2}$$

are the detection and false alarm probabilities at $S2$, respectively, $u_2 = k$ implies that sensor $S2$ decides H_k, $k = 0, 1$, and $\delta(x)$ is Kronecker's delta

$$\delta(x) = \begin{cases} 1 & x = 0 \\ 0 & x \neq 0 \end{cases}$$

Hence, if t is the threshold at sensor $S1$, the test at $S1$ reduces to

$$\frac{p(r_1|H_1)P_{D2}}{p(r_1|H_0)P_{F2}} \underset{H_0}{\overset{H_1}{\gtrless}} t \quad \text{if} \quad u_2 = 1$$

$$\frac{p(r_1|H_1)(1 - P_{D2})}{p(r_1|H_0)(1 - P_{F2})} \underset{H_0}{\overset{H_1}{\gtrless}} t \quad \text{if} \quad u_2 = 0. \tag{A.3}$$

Alternatively

$$\Lambda(r_1) \underset{H_0}{\overset{H_1}{\gtrless}} \begin{matrix} t_{1,1} & \text{if } u_2 = 1 \\ t_{1,0} & \text{if } u_2 = 0 \end{matrix} \tag{A.4}$$

where

$$\Lambda(r_1) = \frac{p(r_1|H_1)}{p(r_2|H_0)}$$

and

$$\frac{t_{1,1}}{t_{1,0}} = \frac{P_{F2}(1 - P_{D2})}{P_{D2}(1 - P_{F2})}. \tag{A.5}$$

References

[1] R. R. Tenney and N. R. Sandell, Jr., "Detection with distributed sensors," *IEEE Trans. Aerospace Electron. Syst.*, vol. AES-17, pp. 501–510, July 1981.

[2] S. C. A. Thomopoulos, R. Viswanathan, and D. K. Bougoulias, "Optimal decision fusion in multiple sensor systems," *IEEE Trans. Aerospace, Electron. Syst.*, vol. AES-23, no. 5, Sept. 1987.

[3] S. C. A. Thomopoulos, D. K. Bougoulias, and L. Zhang, "Optimal and suboptimal distributed decision fusion," *SPIE's 1988 Tech. Symp. Optics, Electro-Optics, Sensors*, Orlando, FL, Apr. 4–8, 1988.

[4] R. Srinivasan, "Distributed radar detection theory," *IEE Proceedings, Part F*, vol. 133, no. 1, pp. 55–60, Feb. 1986.

[5] S. C. A. Thomopoulos, R. Viswanathan, and D. K. Bougoulias, "Optimal distributed decision fusion," *IEEE Trans. Aerospace Electron. Syst.*, vol. 25, no. 5, pp. 761–764, Sept. 1989.

[6] R. Viswanathan, S. C. A. Thomopoulos, and R. Tumuluri, "Optimal serial distributed decision fusion," *IEEE Trans. Aerospace Electron. Syst.*, vol. 24, no. 4, July 1988.

[7] J. D. Papastavrou and M. Athans, "A distributed hypothesis-testing team decision with communications cost," in *Proc. 25th Conf. Decision Contr.*, Athens, Greece, Dec. 1986.

[8] J. D. Papastavrou, "Distributed detection with selective communications," Ph.D. dissertation, Dept. Elect. Eng., M.I.T., Cambridge, MA, 1986.

[9] H. L. Van Trees, *Detection, Estimation and Modulation Theory, Part I*. New York: Wiley, 1968.

[10] J. V. Di Franco and W. L. Rubin, *Radar Detection*. MA: Artech House, 1980.

[11] S. C. A. Thomopoulos, "Sensor integration and data fusion," *J. Robotic Syst.* (Special Issue), vol. 7, no. 3, 1990.

[12] S. C. A. Thomopoulos and N. N. Okello, "Distributed detection with consulting sensors and communication cost: analysis and numerical results," IPIS Lab., Dept. Elect. Eng., SIU, Carbondale, IL, Tech. Rep. TR-SIU-EE-10, 1988.

[13] —, "Distributed detection with consulting sensors and communication cost," *Decision Contr. Syst. Lab.*, Dep. Elect. Comput. Eng., The Pennsylvania State Univ., University Park, PA, Tech. Rep., Oct. 1990.

0018-9286/92$03.00 © 1992 IEEE

SENSOR INTEGRATION AND DATA FUSION

STELIOS C. A. THOMOPOULOS

Reprinted from
Journal of Robotic Systems
Vol. 7 No. 3 June 1990

Sensor Integration and Data Fusion

Stelios C. A. Thomopoulos
Department of Electrical Engineering
The Pennsylvania State University
University Park, PA 16802
Received December 12, 1989; accepted January 23, 1990

The problem of sensor integration and data fusion is addressed. We consider the problem of combining information from diversified sources in a coherent fashion. We assume that information from various sensors may be available in different forms at the fusion. For example, data from infrared (IR) sensors may be combined with range radar (RR) data and further combined with visual images. In each case, data and information from different sensors are presented in a different format which may not be directly compatible for all sensors. Part of the available information may be in the form of attributes and part in the form of dynamical measurements. A generalized evidence processing theory and an architecture for sensor integration and data fusion that accommodates diversified sources of information are presented. Data (or, more generically, information) fusion may take place at different levels, such as the level of dynamics, the level of attributes, and the level of evidence. The common and different aspects of fusion at the different levels are investigated and several practical examples of real world data fusion problems are discussed.

本論文ではセンサ統合とセンサ融合について述べられる。分散した情報源からの情報を等価なものとして統合する問題について考察する。ここでは異なる個々のセンサからの情報は融合の際には異なった形式にあると仮定する。例えば赤外線からのデータがレンジ・レーダと結合されたり、さらに視覚画像と結合されたりする。各々の場合、異なるセンサからのデータや情報はすべてセンサ間で直接変換不可能な異なるフォーマットで表現されている。有効な情報のある部分はアトリビュートの形式になっており、ある部分は動的計測の形式になっている。一般化されたエビデンス計算理論と分散した情報源を調整するデータ融合が述べられる。データ(より一般的には情報)融合は異なるレベルで行われる。例えばダイナミクスのレベル、アトリビュートのレベル、エビデンスのレベルなどである。各々の異なったレベルにおける融合が持つ一般性と特殊性が研究され、実世界のデータ融合問題のいくつかの例が議論される。

INTRODUCTION

Although, for millions of years, nature has provided solutions to sensor integration and data fusion problems in a very successful manner (even low level and low complexity organisms integrate information from different sensory

Journal of Robotic Systems 7(3), 337–372 (1990)
 CCC 0741-2223/90/030337-36$4.00

systems routinely), it is only recently that the fusion problem has been addressed by the scientific community; and for good reasons.

The proliferation of inexpensive sensory devices makes possible the collection of data about a phenomenon or a process from different and diversified sensors operating at different bandwidths of the frequency spectrum, with different resolution, different reliability, and different semantic interpretation. Inexpensive microprocessors have made the processing of data at the local sensor level feasible resulting in the creation of local inference at the sensor level. Raw data and processed information in the form of parameter or state estimates regarding a phenomenon or a process, or in the form of evidence supporting certain propositions or decisions favoring certain hypotheses, is gathered at a central unit, often called the fusion center. This semantically diverse and diversified information must be integrated at the fusion into some form of compound inference that can be used intelligently to yield a better understanding of a phenomenon or a process, improve the decision-making capabilities of a classifier, enhance the recognition capabilities of an image analyzer, increase the precision of a tracker, reinforce the control ability of a controller, etc. Sensor fusion is then the process of integrating raw and processed data into some form of meaningful inference that can be used intelligently to improve the performance of the system, measured in any convenient and quantifiable way, beyond the level that any one of the components of the system separately or any subset of the system components partially combined could achieve.

Sensor fusion differs from signal processing. In sensor fusion data and information may be presented differently and combined at different levels, from raw data, where dynamics are taken into consideration, to evidence, when estimates are available, to attributes, when classification has already taken place at a local level. In signal processing, a precise mathematical model that describes the generation of the data is assumed to be available. In sensor fusion, however, it is not always true that a precise mathematical model for the data generation exists or can be derived. At times, not even a simple mathematical model may be available. Under these circumstances, conventional signal processing techniques may be completely inadequate for sensor fusion. Signal processing may be thought of as only the first stage of processing in sensor fusion. It is the diversified information processing aspect that differentiates sensor fusion from signal processing. Processing that may be done incoherently (i.e., asynchronously) and often lacks mathematical formulation. Sensor fusion can be thought of as a two-stage processing with a signal preprocessor unit at the lowest level and an information, or evidential, combiner at the highest level.

The objective of this article is to provide an analytical framework for evidence processing and a generic architecture for sensor fusion that will allow the integration of data and processed information of different format into coherent inference or intelligence. Since the objective of sensor fusion is the improvement of the performance of a system after fusion beyond that of any of its components or subset of its components separately before fusion, the analytical framework and the generic architecture should meet this objective. The selection of criteria against which the performance of a fusion system is measured is left up to the

particular task that the fusion system is designed to perform. It should be noticed, however, that what is required from the performance of the system after fusion is only improvement and not optimality, since optimality may not even be definable for a sensor fusion problem in the absence of a mathematical model.

At this point, it is legitimate to ask whether a systematic approach for sensor fusion exists, consistent with the definition given earlier, that will allow the development of new and powerful techniques resulting in: (1) a deeper understanding of the sensor fusion problem; and (2) more comprehensive solutions to the inference making problem with diversified sources of information. The objective of this article is to provide a unified sensor theory along these lines and demonstrate that the fusion architecture is consistent with the objectives of sensor fusion set above.

The paper is structured as follows. In the first section, a generic sensor fusion architecture along with several fusion system design criteria are introduced. The architecture consists of three distinct levels of processing. The second through fourth sections elaborate on the different fusion levels of the proposed architecture. A new generalized evidence processing theory is introduced in the third section. The theoretical concepts developed in the first three sections are applied in the fifth section in a concrete paradigm of a fusion system for object tracking consisting of stereo cameras, pulse radar(s), and range radar(s). The article concludes with a look to future directions and problems in sensor fusion.

DATA FUSION ARCHITECTURE AND DESIGN CRITERIA

In the fusion architecture that is proposed, sensor integration is accomplished at three different levels: the signal level, the level of evidence, and the level of dynamics.

At the signal level, sensor integration and data fusion takes place through correlation and learning. A typical characteristic of fusion at this level is, in general, the lack of a convenient mathematical model that describes the phenomenon or the process that can lead to analytical solutions. At this level, sensor integration can be accomplished through heuristic rules, correlation, or by using trainable networks, such as an artificial neural network, that are designed and trained so that they take advantage of particular, probably nonquantifiable, properties of the measured signals at the sensors.

At the evidence level, a statistical model describing the observed phenomenon or process is required. The statistical model need not necessarily be precise. Fuzziness in the model may also be included to allow for incomplete and inconclusive evidence. At the evidence level, the sensors collect data from the observed phenomenon or process, and process it along with other information that may have been collected from other sensors depending on the sensors topology and communication network. The outcome of this processing in the form of local inference from each sensor is relayed to the fusion. The fusion combines this local inference into a final inference in a form that can be used for decision making, identification, hypothesis testing, control, etc.

At the level of dynamics it is usually assumed that a mathematical model that describes the process from which data is collected using multiple sensors exists. Furthermore, it is assumed that the data is some known transformation, linear or nonlinear, of the process state(s). Thus, the data can be fused either in a centralized fashion by combining the observations and then processing them as a whole, or in a decentralized fashion, by either having each sensor individually process its own data and then merging the processed data at the fusion, or by grouping the sensors into groups, having each group process all the data from the sensors in the group, and then combining the processed data at the fusion.

The proposed integrated fusion architecture consists of three different modules, one for each of the three fusion levels described above, Figure 1. The architecture in Figure 1 suggests a sequential and rather hierarchical processing of information from the signal level down to the evidential level, to the level of dynamics. However, this need not always be the case. Processing at the different level may take place concurrently and in parallel. Even more, the processing ordering in the fusion architecture can be completely reversed depending on the application. Hence, the interconnections among the three modules in Figure 1 are more indicative than absolute, and they definitely depend on the particular fusion application.

In all cases, data combining at the fusion is done according to an objective that is set a priori. For example, if tracking with multiple sensors is the objective, the end result of the fusion must be a target position and velocity estimates. In this case, the data may come from diversified sensors such as infrared radar,

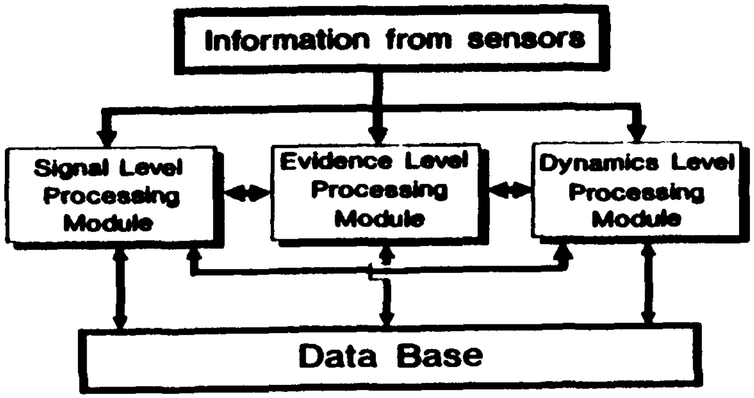


Figure 1. Sensor integration and data fusion architecture

range radar, ladar, imagining systems, etc. If in addition, the presence of the target is ambiguous and the type of the target itself is not known, detection and identification may precede the estimation. A question associated with sensor integration and data fusion that is often asked, sometimes in the spirit of criticism, is whether it makes sense to fuse data from different sensors, or one is better off by using data from a single sensor. In other words, can the performance of a system, such as a tracker, a detector, an identifier, or a controller, be improved after fusion compared to what it would have been if a single sensor were used? An answer to this question may not always be feasible and definitely depends on the reliability of each sensor and on the method that is used to fuse the data from the different sensors together. Theoretical and numerical results that support data fusion can be found in Refs. 1–6, 11, and 17.

Data fusion results in Tables I, II, and III, indicate that the detection probability after fusion can be higher than the highest detection probability of any of the peripheral sensors in the fusion system while the false alarm at the fusion is kept lower than the lowest false alarm probability of any one of the sensors in the fusion system (Theorem 1[1]). The results in these three tables were obtained for a parallel sensor topology.[1] (Fig. 2), with sensors relaying conditionally independent, binary decisions to the fusion. Each sensor was set to operate at fixed signal-to-noise ratio and fixed false alarm probability in a slow fading Rayleigh channel. The peripheral binary decisions from the sensors were combined at the fusion using a Neyman–Pearson test. Similar results are obtained for a serial sensor topology (Fig. 3).[2] The results suggest that data fusion can be used to improve the performance of a system beyond what each component, or subset of components, is capable of achieving. Furthermore, use of the proper inference format can improve the fusion performance significantly. Tables IV and V are indicative of the improvement in the performance of a parallel sensor topology fusion when a single additional bit of quality information is attached to the sensors binary decisions indicating the confidence of each sensor on its decision.[1,17] A region of low confidence was associated with data falling within 10% on either side of the threshold that would have been used to decide in favor of one of the two tested hypothesis according to a *prespecified false alarm probability*. If the data fell outside the low confidence region, a high degree of confidence was associated with the corresponding decision.

Thus, numerical results indicate that sensor fusion can improve the performance of a system beyond what any of the components or a subset of the components can achieve in agreement with the objectives set above. However, for a successful sensor integration and data fusion system, it is imperative that the design of the fusion be based on criteria that are set a priori to guarantee that the performance of the fusion system will be superior to any of its components or subset of its components alone. Our experience in data fusion has been distilled to three design criteria which we consider essential for any data fusion system of practical significance.[18,19] The *three design critia* are:

(1) Monotonicity with respect to fused information.

Table I. Fusion system of five sensors in parallel topology. All sensors have the same false alarm probability $P_{F_i} = 0.05$, $i = 1, \ldots, 5$. The corresponding detection probabilities are as follows.[a]

i	1	2	3	4	5
P_{D_i}	0.95	0.94	0.93	0.92	0.91

Decision fusion: 5 — Sensor system
Sensors PF: Equal x — Unequal -
Sensors PD: Equal - — Unequal x

Threshold @ fusion center	Probability of detection @ fusion center	Probability of false alarm @ fusion center
PDMAX = 0.95000	PFMIN = 0.50000E-01	
t^*	PD	PF
6163.2	0.957817	0.388000E-04
53.004	0.963797	0.142812E-03
45.880	0.968973	0.255625E-03
40.339	0.973523	0.368437E-03
38.907	0.977913	0.481250E-03
34.208	0.981772	0.594062E-03
32.081	0.985391	0.706874E-03
29.610	0.988731	0.819687E-03
28.207	0.991913	0.932499E-03
24.416	0.994668	0.104531E-02
20.705	0.997003	0.115812E-02
0.20998	0.997454	0.330156E-02
0.17806	0.997835	0.544500E-02
0.15413	0.998165	0.758843E-02
0.14683	0.998480	0.973187E-02
0.13552	0.998771	0.118753E-01
0.12709	0.999043	0.140187E-01
0.11174	0.999282	0.161622E-01
0.10778	0.999513	0.183056E-01
0.94760E-01	0.999717	0.204490E-01
0.82023E-01	0.999892	0.225925E-01

[a] After fusion, the detection probability is higher than the highest detection probability among all sensors for a wide range of false alarm probabilities lower than 0.05.

(2) Monotonicity with respect to the cost associated with acquiring the information.
(3) Robustness with respect to unknown priors and, in general, any a priori uncertainty.

In case that robustness is not achievable, some form of monotonicity of the performance measure with respect to the a priori uncertainty should be required.

The three design criteria are intuitively pleasing. Furthermore, if a fusion

Table II. Fusion system of five dissimilar sensors in parallel topology. The sensors false alarm probabilities P_{F_i} and detection probabilities P_{D_i}, $i = 1, \ldots, 5$, are as follows.[a]

i	1	2	3	4	5
P_{F_i}	0.05	0.04	0.03	0.02	0.01
P_{D_i}	0.95	0.94	0.93	0.92	0.91

Decision fusion: 5 — Sensor system
Sensors PF: Equal x — Unequal -
Sensors PD: Equal - — Unequal x

Threshold @ fusion center	Probability of detection @ fusion center	Probability of false alarm @ fusion center
PDMAX = 0.95000	PFMIN = 0.10000E-01	
t^*	PD	PF
57882.0	0.957817	0.269200E-05
426.86	0.960153	0.816400E-05
373.63	0.962908	0.155360E-04
358.72	0.966248	0.248400E-04
284.83	0.969430	0.360200E-04
273.46	0.973209	0.501320E-04
239.36	0.977840	0.691439E-04
160.34	0.981459	0.917159E-04
153.94	0.985848	0.120228E-03
134.74	0.991024	0.158640E-03
102.72	0.997003	0.216852E-03
0.99369	0.997179	0.393700E-03
0.75752	0.997382	0.661908E-03
0.66305	0.997622	0.102314E-02
0.63660	0.997912	0.147942E-02
0.42643	0.998143	0.202115E-02
0.37325	0.998416	0.275098E-02
0.35836	0.998746	0.367287E-02
0.28454	0.999061	0.477889E-02
0.27319	0.999442	0.617598E-02
0.23912	0.999892	0.805816E-02
1 Sensor off		
PDMAX = 0.95000	PFMIN = 0.20000E-01	
t^*	PD	PF
1129.9	0.976981	0.150400E-03
4.6908	0.979548	0.697600E-03
4.1058	0.982575	0.143480E-02
3.9420	0.986246	0.236600E-02
3.1300	0.989742	0.348320E-02
3.0051	0.993983	0.489440E-02
2.6303	0.996984	0.679560E-02
2 Sensors off		
PDMAX = 0.95000	PFMIN = [illegible]	
t^*	PD	PF
32.222	0.999720	[illegible]

[a] After fusion, the detection probability is higher than the highest detection probability among all sensors for a wide range of false alarm probabilities lower than the lowest false alarm probability among the five sensors.

Table III. Fusion system of four similar sensors in parallel topology. All sensors operate at the same false alarm probability and detection probability level $(P_{F_i}, P_{D_i}) = (0.05, 0.95)$, $i = 1, \ldots, 5$. A single additional quality information bit is added to their decisions before they are transmitted to the fusion. The data that follows is Gaussian distributed with quality bit coefficients.[a]

C_{ij}	H_1	H_0
C_{11}	0.948	0.46
C_{01}	0.052	0.54
C_{00}	0.52	0.047
C_{10}	0.48	0.953

Decision fusion: 4	Sensor system with quality bits
Sensors PF: Equal x	Unequal –
Sensors PD: Equal x	Unequal –

Threshold @ fusion center	Probability of detection @ fusion center	Probability of false alarm @ fusion center
PDMAX = 0.95000	PFMIN = 0.50000E-01	
t^*	PD	PF
62318.0	0.956002	0.175551E-05
20357.0	0.960930	0.199000E-05
9390.7	0.961918	0.210220E-05
2988.7	0.963462	0.261876E-05
2911.9	0.980782	0.856706E-05
951.21	0.981595	0.942131E-05
926.74	0.990711	0.192580E-04
438.79	0.990738	0.193191E-04
302.74	0.990880	0.197900E-04
139.65	0.990937	0.201943E-04
136.06	0.992362	0.306685E-04
44.446	0.992406	0.316713E-04
43.303	0.993906	0.663133E-04
42.189	0.998114	0.166041E-03
20.503	0.998114	0.166055E-03
14.146	0.998129	0.167161E-03
13.782	0.998524	0.195805E-03
6.5253	0.998525	0.195924E-03
6.3575	0.998577	0.204121E-03
4.5021	0.998579	0.204578E-03
2.0768	0.998580	0.204970E-03
2.0234	0.998662	0.245637E-03
1.9713	0.999354	0.596850E-03
0.66097	0.999355	0.597499E-03
0.64397	0.999398	0.664750E-03
0.62741	0.999762	0.124555E-02
0.29706	0.999763	0.124796E-02
0.21036	0.999763	0.124850E-02
0.20495	0.999771	0.128557E-02
0.945441E-01	0.999772	0.130148E-02
0.92113E-01	0.999810	0.171378E-02
0.66951E-01	0.999810	0.171395E-02
0.30090E-01	0.999811	0.175343E-02
0.29316E-01	0.999851	0.311705E-02

Table III. (Cont)

[a]After fusion, the detection probability is higher than the highest detection probability among all sensors for a wide range of false alarm probabilities lower than the lowest false alarm probability among the five sensors. The additional quality bit information improves the performance significantly.

Table III. (Cont) Comparative results from three different fusion systems with four ($N = 4$) sensors, all operating at level $(P_F, P_D) = (0.05, 0.95)$ when the individual sensors transmit

	P_f^f	P_D^f
Only decisions	0.014	0.9995
Decision with one quality bit	0.0013	0.9998
Raw data (best centralized N-P test)	0.001	0.9998

system is designed according to these criteria, its performance will be superior to any of, or any subset of, its components. The three criteria were used in the design of a fusion system consisting of two consulting sensors in the presence of communication cost.[18] Several fusion schemes that satisfied different optimality criteria were designed according to these three criteria.[18,19]

In addition to the three design criteria, several other communication aspects related to networks of distributed sensors need to be taken into consideration in the design of a data fusion. In a network of geographically distributed sensors,

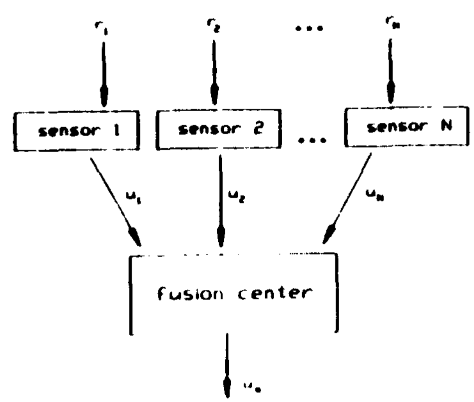


Figure 2. Parallel sensor topology

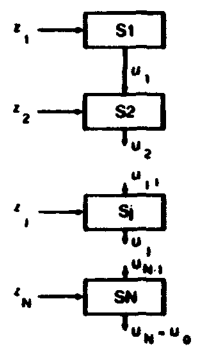


Figure 3. Serial sensor topology.

communication aspects, like delays associated with the transmission of data or information from sensor-to-fusion and sensor-to-sensor, and channel errors, must be taken into account in the design of the fusion. The optimal fusion in the presence of delays in the transmission of information from the sensors to the fusion and errors due to noisy communication channels was designed in [20] for the parallel sensor topology. Figures 4, 5 and 6 show how communi-

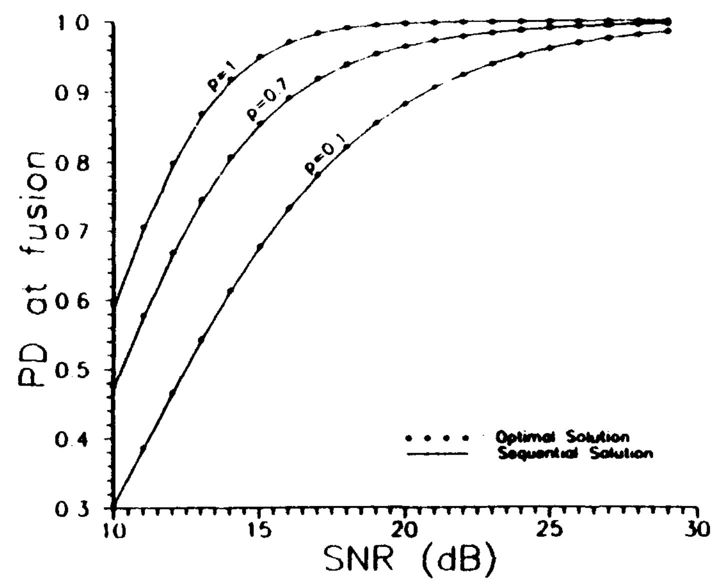
Table IV. Fusion system of two similar sensors in parallel topology. Both sensors operate at the same false alarm probability and detection probability level $(P_{F_i}, P_{D_i}) = (0.05, 0.95)$, $i = 1, \ldots, 5$. A single additional quality information bit is added to their decisions before they are transmitted to the fusion. The quality bit coefficients are the same as in Table III. After fusion, the detection probability increased beyond that of each sensor individually for false alarm probability lower than the sensors false alarm probability, which is not feasible without the use of quality bit information (Theorem 1[1]). The additional quality bit information improves the performance significantly.

Decision fusion: 2 Sensor system with quality bits
Sensors PF: Equal x Unequal -
Sensors PD: Equal x Unequal -

Threshold @ fusion center	Probability of detection @ fusion center	Probability of false alarm @ fusion center
PDMAX = 0.9500	PFMIN = 0.5000E-01	
t^*	PD	PF
1.0654	0.951900	0.696499E-02
1.0380	0.995129	0.486111E-01

cation delays and channel errors affect the design of the optimal fusion, and thus, need to be taken into consideration in determining the fusion and sensors operating points. The numerical results in these three figures were obtained for the parallel sensor topology by assuming that the binary decisions might be encountered by delays and not be available at the fusion at the time of fusion. Thus, the decisions from each sensor were assumed to be available for fusion with certain probability p. From Figures 4 and 5, the effect of probability p on the detection probability at the fusion is apparent. Furthermore, the effect of the probability of error in receiving a sensor decision incorrectly is shown in Figure 6. It turns out that the error probability not only does it affect the fusion performance but it also imposes a limit on the achievable false alarm at the fusion.[20]

When fusion taken place at the level of dynamics, the daley factor plays a significant role in distributed estimation and control where its effect is often overlooked. The problem of estimation and control with distributed sensors in



Comparison of two solutions for N=3 and equal SNR case under OR policy

Figure 4. Probability of detection after fusion for different delays. Decision fusion with three similar sensors, with equal signal to noise ratio, in parallel topology. The parameter p indicates the probability that a decision from a sensor is present at the fusion during the fusion period. The sensor decisions are fused according to the OR rule.

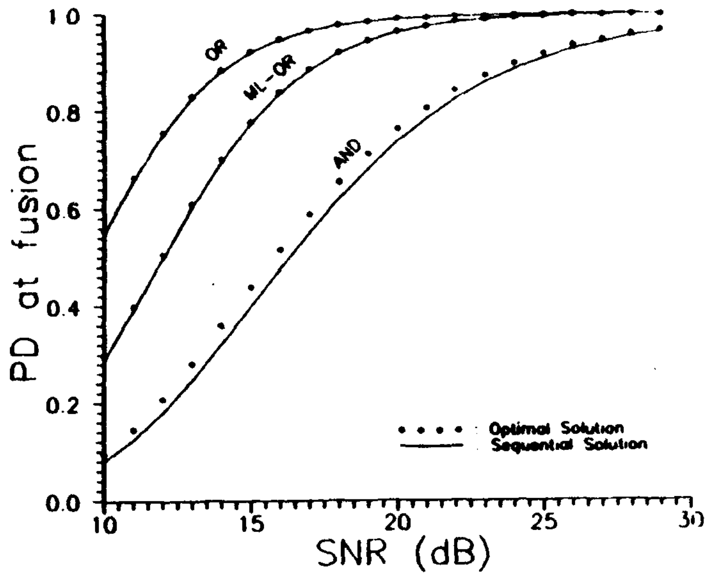


Figure 5. Probability of detection after fusion for three different fusion rules and delay 0.9. Decision fusion with three similar sensors in parallel topology. The fusion rules that were used to fuse the sensor decisions were the OR, the AND, and the Majority-OR rule (ML-OR). In the ML-OR rule, decision fusion is done using the majority rule when all the decisions are present at the fusion, otherwise the OR rule is used.[20] The OR policy is the optimal fusion rule.[20]

the presence of delays in the transmission of both information and control signals has been considered in Refs. 21–24, where the minimum mean squared error filter and the optimal quadratic cost controller were derived in the presence of transmission delay and uncertainty in the sampling times.

Another point that needs to be addressed in sensor fusion is the possible mismatch among different sensors.[26] In various circumstances, either due to geographical disparity or misalignment, different sensors may not gather data from precisely the same geographical region, or may not observe the same phenomenon or process simultaneously. Hence, a mismatch error may occur which need to be considered in the design of the fusion. The difficulty arises when the mismatch cannot be resolved from the data collected from the different sensors. In this case it is necessary that the mismatch is modeled mathematically, if possible. The model can then be incorporated in the design of the fusion. However, it is not always possible to identify a suitable mathematical model to describe sensor mismatch. When a mathematical model for the sensor mismatch is not available, data fusion should take place at the signal level by taking into consideration the constraints imposed by the problem and sensor physics. This is for example the case in stereo image registration.[26,27] A mathematical model for sensor mismatch, particularly useful in radar data fusion and pixel registration in stereo vision, was introduced in Ref. 25 along with different fusion designs. The reader is referred to Ref. 25 for some interesting and surprising results in sensor fusion with sensor mismatch.

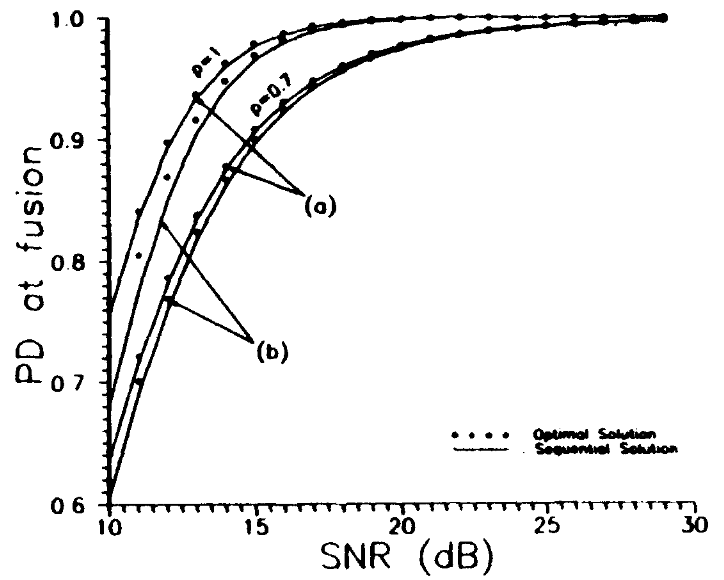


Figure 6. Probability of detection after fusion for different delays and channel errors. Decision fusion with three similar sensors in parallel topology. The parameter p indicates the probability that a decision from a sensor is present at the fusion during the fusion period. For the set (a) curves, the channel error probability is $P_c = 0.00001$. For the set (b) of curves, the error probability is $P_c = 0.00003$. The decisions are fused according to the OR rule. The effect of the channel errors is apparent.[20]

DATA FUSION AT THE SIGNAL LEVEL

A typical example of sensor fusion at the signal level occurs in image processing. One of the tough problems in stereo vision is the pixel registration between a left and a right image. In order to reconstruct depth from stereo (images), the pixels that correspond to the same point on the object must be first identified.[27,34] *Since the transformation from the object to the left and right images* (i.e., the projection from the 3-D space of the object onto the 2-D spaces of the right and left images) does not maintain this information, this information is not retrievable from geometrical modeling and explicit mathematical inversion. For *a given pixel on the left image that is turned on (dark spot in Figure 7)*, there is a multiplicity of pixels on the right image that can correspond to the same point on the object. As a matter of fact, all the intersection points between the line of sight associated with a turned-on pixel on the left image with all the lines of sight that correspond to pixels that are turned on on the right image qualify as legitimate associations. In Figure 7, a possible registration among pixels from the two images is shown. In Figure 7, a dark pixel indicates that an object is present, whereas a light pixel indicates the absence of an object. All intersections among lines of sight from pixels that are "on" correspond to legitimate registrations. The particular registration that is shown in Figure 7 (blacken intersection points) corresponds to a flat object (identical depth). Since a mathematical model for fusing the information from the right and left images in order to generate proper registrations is not available, an alternative way may be used to fuse the information based on properties and constraints imposed by the physics of the problem.

A neural-type algorithm that is based on excitatory and inhibitory connections (synapses) among the pixels imposed by the physics of the problem has been successfully used for stereo image registration.[26] Besides the fact that no analytical model is always required to describe the phenomenon or the process that generates the data, an additional advantage of the neural approach is the parallel processing capabilities which allow for large amount of data included in raw signals to be processed efficiently.

The matching algorithm is based on physical constraints imposed by the *assumed relative smoothness of the object surface. The assumptions the fusion*

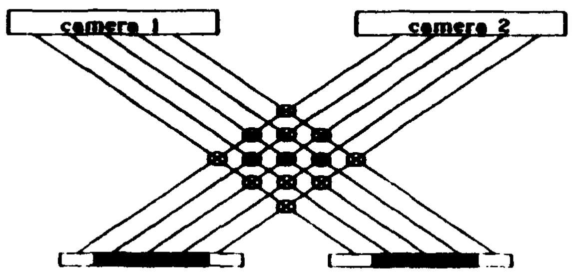


Figure 7. Stereo camera image fusion

algorithm is based on are:

(1) Each point in an image can have one depth value only
(2) A point is very likely to have a depth value near the values of its neighbors

The same two assumptions were first used for *image registration* by Marr and Poggio.[28] The algorithm in Ref. 26 resembles that of Ref. 28. The algorithm is recursive in nature and converges when all pixels on one image have been associated with pixels on the other image. It is given by

$$C_{n+1}(x,y,d) = \sigma \sum_{x',y',d' \in S} C_n(x',y',d') - \epsilon \sum_{x',y',d' \in O} C_n(x',y',d') + \beta C_0(x',y',d') \quad (1)$$

where S corresponds to the excitatory connections and O corresponds to the inhibitory connections among pixels. The constants σ, ϵ, and β are chosen to guarantee the best match. The function C is a sigmoid function and is given the value one if a certain threshold is exceeded and zero if otherwise. The third term in (1) is a bias that is introduced and may reflect additional information included in the initial registration. By choosing the excitatory and inhibitory regions S and O properly, stereo images from objects with different degrees of smoothness can be *registered*. Notice that the algorithm (3) can be implemented entirely in parallel. A neural network equivalent schematic implementation is shown in Figure 8. The circles in Figure 8 correspond to neurons and are

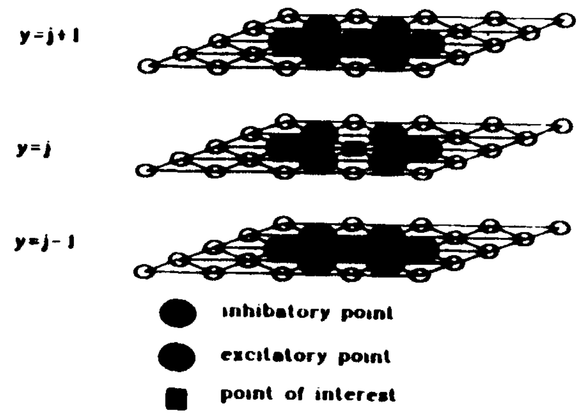


Figure 8. Artificial neural network for stereo image fusion

associated with the intersections of the lines-of-sight from the pixels of the right and left cameras. Each grid in Figure 8 represents a layer in the neural network. Synapses (i.e., interconnections) among the neurons are among the six closest neighbors on the same layer, and on one layer above and one layer below. The eight excitatory connections correspond to neurons that correspond to points that have the same depth as the point of interest. If the neurons with the excitatory interconnection are on, they tend to reinforce the depth that corresponds to the point of interest. However, if the neurons with inhibitory connections are on, they tend to keep the point of interest off, since only one depth value can be assigned to a point. A depth map that resulted from the registration of two stereo images using the described ANN is shown in Figure 9 along with the unregistered stereo images. A "1" in the stereo image indicate that the pixel "on," whereas a "0" indicates that the pixel is off. The value in the depth map indicate depth. An interpolation algorithm can be used to smooth out the depth values.

Once the matching coordinate pairs from the left and right image have been found, the depth can be retrieved from geometrical relationships that hold in perspective projection by

$$z = f - \frac{2df}{x'' - x'} \tag{2}$$

where z is the depth, (x'', x') is the corresponding right and left coordinate pixel pair, f is the camera focal length, and $2d$ is the distance between the two cameras in the direction of the x-axis. For a more detailed description of the registration algorithm and applications of the algorithm on tracking 3-D moving objects see Ref. 35. In Ref. 35 a detailed analysis of the fusion of stereo images and range radar data for tracking objects sustaining 3-D translational and rotational motion is given. Simulation results in Refs. 35 and 36 indicate that fusion of information as diversified as stereo images and range radar measurements improves the tracking accuracy considerably.

In the pixel registration fusion problem, the statistical model that was introduced in Ref. 25 can be applied to model the mismatch mathematically. However, certain geometrical assumptions that are camera position dependent need to be made first, thus making the modeling somewhat artificial. Hence, the fusion approach that is based on physical constraints imposed by the physics of the problem and direct processing of the data at the signal level alleviates the artificial geometrical assumptions required in the mathematical modeling of the mismatch. However, no matter what the approach, information mismatch is an important aspect of sensor fusion. From our experience, it seems that in the presence of mismatch, information fusion is performed more efficiently at the signal level if a naturally arising mathematical model to describe the mismatch is not available.

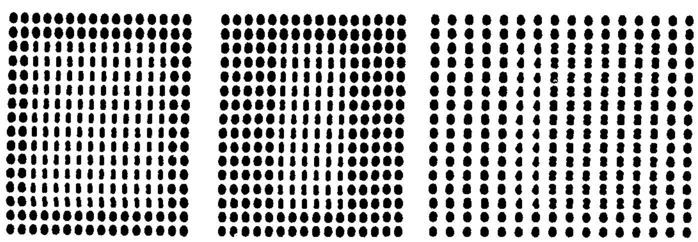


Figure 9. Initial images and resulting depth values

DATA FUSION AT THE EVIDENCIAL LEVEL

Two major evidence combining theories have dominated the field of distributed evidence processing: the Bayesian theory[4,5,12,16] and the Dempster–Shafer's (D-S's) theory.[7,10] Both theories have advantages and disadvantages. Proponents of the D-S theory criticize the Bayesian theory for lack of flexibility when it comes to fuzzy decisions, i.e., when the available evidence does not support hard decisions. On the other hand, proponents of the Bayesian approach criticize the D-S theory for lack of rigorousness in the axiomatic definition of evidence through independent experiments. In this article we present a new theory that unifies the Bayesian with the D-S theory. The *Generalized Evidence Processing* theory that is presented in this paper, and was introduced in Ref. 29, combines the advantages of both theories without, we believe, any of their disadvantages. To simplify our presentation, we only consider the binary hypothesis testing case throughout this paper. Generalization to the multiple hypothesis cases can be done routinely. The mathematical expressions become more complicated, yet the structure of the theories remains the same. The new *Generalized Evidence Processing* (GEP) theory unifies the Bayesian theory with the Dempster–Shafer theory in a general framework.

Generalized Evidence Processing Theory

Let H_1, H_0 be the two hypotheses under test. The probability space is partitioned into two regions according to the events $\{\omega = H_1\}$ and $\{\omega = H_0\}$ with associated probabilities $P_1 \geq 0$ and $P_0 \geq 0$ respectively, where $P_1 + P_0 = 1$.

Let d_0, d_1, and $d_{0\vee1}$ be a frame of discernment used by a decision maker to partition the probability space according to the gathered evidence, where the three decisions correspond to the propositions "H_0 true," "H_1 true," and "H_0 or H_1 true," respectively. The decision $d_{0\vee1}$, where "$\vee$" stands for "*or*," indicates the inability of the decision maker to come up with conclusive evidence on the true nature of the hypothesis.

In the classical probabilistic (Bayesian) framework, the probability associated with $d_{0\vee1}$ is equal to

$$Pr\{d_{0\vee1}\} = Pr\{H_0 \vee H_1\} = Pr\{H_0\} + Pr\{H_1\} = 1 \tag{3}$$

since H_0 and H_1 constitute a disjoint coverage of the probability space over which the evidence processing problem is defined. As it was mentioned earlier, the apparent weakness of the Bayesian theory to incorporate nonmutually exclusive, i.e., redundant, propositions gave rise to the D-S theory, which is particularly efficient in dealing with fuzzy propositions. However, an extended unified framework can be used to create a unified evidence theory that includes both the Bayesian and D–S theory as special cases and is suitable for sensor fusion systems design.

Let z be the transformation from the initial event space Ω into the observation (data) space $\mathbf{Z}$, i.e.,

$$z: \Omega \to \mathbf{Z} \tag{4}$$

and d be a transformation from the observation space into the decision space $\mathbf{D}$, i.e.,

$$d: \mathbf{Z} \to \mathbf{D} \tag{5}$$

Let $\{dP(z|H_j), P(H_j); i = 0, 1\}$ be the probability measure on $\mathbf{Z}$. Let C_{ij} be the cost associated with a decision i when the true hypothesis is H_j. Define the mapping d so that the cumulative risk

$$\mathbf{R} = \sum_i \sum_j C_{ij} P_j \int_{Z_i} dP(z|H_j), \quad j = 0, 1 \quad \text{and} \quad i = 0, 1, 2 \tag{6}$$

is minimized, where $d_2 = d_{0 \cup 1}$ is the ambiguous decision. In (6), the regions Z_i indicate the partition of the observation space according to the decision rule d, where Z_i indicates the region of the observation space in which the decision is d_i. Notice that the partition of the observation space is made according to the set of decisions that is chosen a priori, and that in a given set a particular decision may not favor a specific hypothesis exclusively, opposite to what is customary in the Bayesian theory. However, it is not clear from (6) if such a partition is feasible, since the decision rule is defined through R which depends on the partition $\{Z_i\}$. We show next that for a proper choice of the costs C_{ij}, there exist partitions $\{Z_i\}$ that define legitimate probability measures on $\mathbf{D}$, and thus a generalized decision rule d.

Rewriting Eq. (6) as

$$\begin{aligned} \mathbf{R} = &\int_{Z_0} [P_0 C_{00} dP(z|H_0) + P_1 C_{01} dP(z|H_1)] \\ &+ \int_{Z_1} [P_0 C_{10} dP(z|H_0) + P_1 C_{11} dP(z|H_1)] \\ &+ \int_{Z_2} [P_0 C_{20} dP(z|H_0) + P_1 C_{21} dP(z|H_1)] \end{aligned} \tag{7}$$

the total risk is minimized if the decision rule assigns z (the observation) to the region that corresponds to the least integrand under the three integrals in (7). Hence, the decision rule becomes

$$P_0 C_{00} dP(z|H_0) + P_1 C_{01} dP(z|H_1) \underset{Z_0 \text{ or } Z_2}{\overset{Z_1 \text{ or } Z_2}{\gtrless}} P_0 C_{10} dP(z|H_0) + P_1 C_{11} dP(z|H_1) \tag{8}$$

and symmetrically for the other alternatives. Dividing both sides of (8) by $dP(z|H_0)$ and defining $\Lambda(z) = dP(z|H_1)/dP(z|H_0)$, the decision rule becomes

$$[C_{01} - C_{11}]\Lambda(z) \underset{d_0 \text{ or } d_2}{\overset{d_1 \text{ or } d_2}{\gtrless}} \frac{P_0}{P_1}[C_{10} - C_{00}] \tag{9}$$

where $d_2 = d_{0 \cup 1}$. Similarly,

$$[C_{01} - C_{21}]\Lambda(z) \underset{d_0 \text{ or } d_1}{\overset{d_2 \text{ or } d_1}{\gtrless}} \frac{P_0}{P_1}[C_{20} - C_{00}] \tag{10}$$

and

$$[C_{21} - C_{11}]\Lambda(z) \underset{d_2 \text{ or } d_0}{\overset{d_1 \text{ or } d_0}{\gtrless}} \frac{P_0}{P_1}[C_{10} - C_{20}] \tag{11}$$

From (9)–(11), it is seen that the decision rule depends on the relative values of the C_{ij} costs. We examine three different cases to illustrate the significance of the C_{ij}'s.

Case 1

The associated cost for correct decision is zero, i.e., $C_{11} = C_{00} = 0$, while the cost of incorrect guessing is higher than the cost associated with indecision under both hypotheses, i.e., $C_{ij} > C_{2j}$ for every $i \neq 2$ and $j = 0$ or 1. Under these conditions, the decision rule becomes

$$\Lambda(z) \underset{d_0 \text{ or } d_2}{\overset{d_1 \text{ or } d_2}{\gtrless}} \frac{P_0 C_{10}}{P_1 C_{01}} \tag{12}$$

Similarly,

$$\Lambda(z) \underset{d_0 \text{ or } d_1}{\overset{d_2 \text{ or } d_1}{\gtrless}} \frac{P_0}{P_1} \frac{C_{20}}{C_{01} - C_{21}} \tag{13}$$

and

$$\Lambda(z) \underset{d_2 \text{ or } d_0}{\overset{d_1 \text{ or } d_0}{\gtrless}} \frac{P_0}{P_1} \frac{C_{10} - C_{20}}{C_{21}} \tag{14}$$

In this case, it is seen from (12) through (14) that the optimal test (decision rule d) is of the likelihood type with the indecision region dependent on the relative values of C_{ij}. Different numerical applications are considered next to further illustrate the significance of the C_{ij}'s.

Case 1.1. $C_{00} = C_{11} = 0$, $C_{10} = C_{01} = 1$, and $C_{20} = C_{21} = 1/3$. In this case $C_{10}/C_{01} = 1$, $C_{20}/(C_{01} - C_{21}) = 0.5$, and $(C_{10} - C_{20})/C_{21} = 2$. The partition of the LR by the decision rule in this case is shown in Figure 10. From this figure, it is seen that in this case the indecision region lies between the two definite decision regions, which corresponds to the way that we would intuitively have picked the indecision (uncertainty) region relative to an LRT.[1]

Case 1.2. $C_{00} = C_{11} = 0$, $C_{10} = C_{01} = 1$, and $C_{20} = C_{21} = 0.5$. In this case $C_{10}/C_{01} = C_{20}/(C_{01} - C_{21}) = (C_{10} - C_{20})/C_{21} = 1$, i.e., all three thresholds are the same, and the indecision region is completely eliminated. The partition of the LR by the decision rule is shown in Figure 11. It corresponds to a standard binary hypothesis–binary decision Bayesian problem.

Case 1.3. $C_{00} = C_{11} = 0$, $C_{10} = C_{01} = 1$, and $C_{20} = C_{21} = 2/3$. In this case $C_{10}/C_{01} = 1$, $C_{20}/(C_{01} - C_{21}) = 2$, and $(C_{10} - C_{20})/C_{21} = 1/2$. The partition of the LR by the decision rule in this case is shown in Figure 12. In this case the two definite (hard) decision regions are sandwiched between the two indecision regions opposite to what one would intuitively have defined as an indecision region in an LRT, and exactly opposite to Case 1.1.

Case 2

$C_{00} = C_{11} = 0$, $C_{01} = C_{21}$. In this case $C_{10}/C_{01} = 1$, $C_{20}/(C_{01} - C_{21}) = \infty$, and $(C_{10} - C_{20})/C_{21} = \alpha$. The partition of the LR into different decision regions is shown in Figures 13 and 14, depending on the value of α. In Figure 13, $\alpha > 1$, and the decision d_0 is completely eliminated. In Figure 14 where $\alpha < 1$, there are three decision regions; d_1, d_0, and not d_1. According to the definition of

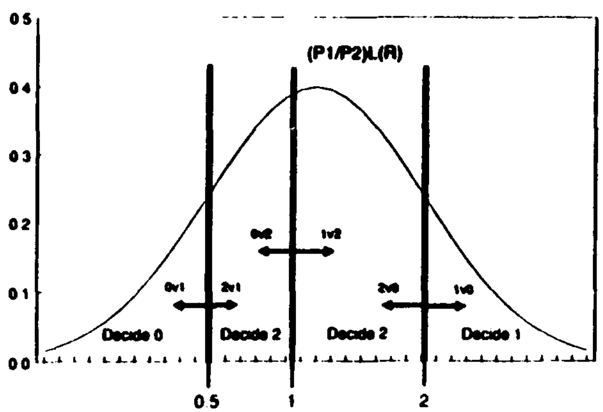


Figure 10. Case 1.1. The indecision region lies between the two definite decision regions.

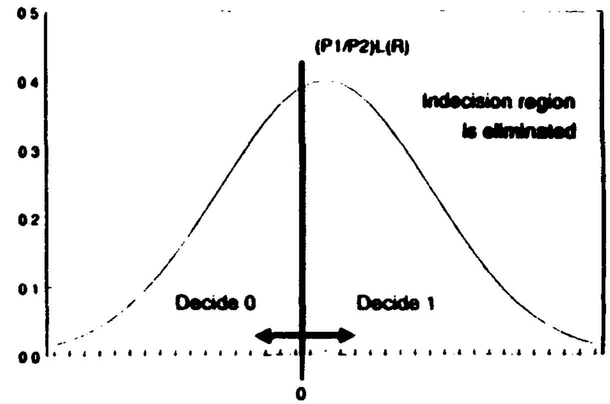


Figure 11. Case 1.2. The indecision region is completely eliminated.

the three possible decisions, the decision "do not decide d_1" must be interpreted as the fuzzy decision "d_0 or d_2" and not as "decide in favor of H_0."

Case 3

If in the above cases $C_{2j} > C_{ij}$, $i \neq 2$ and $j = 0, 1$, the LRT in (13) is reversed and the threshold in (14) becomes negative. Under these circumstances, the decision regions in the previous cases are reversed.

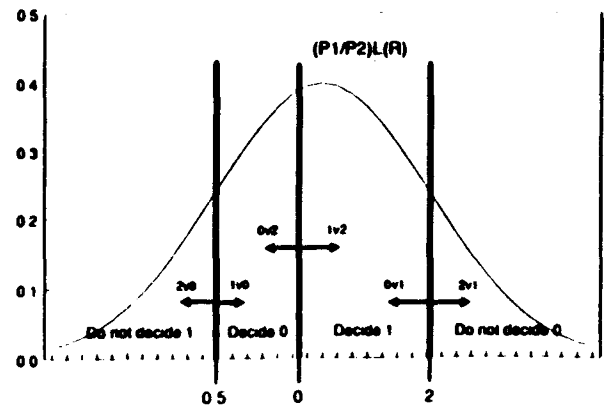


Figure 12. Case 1.3. The definite decision regions lie between the indecision regions.

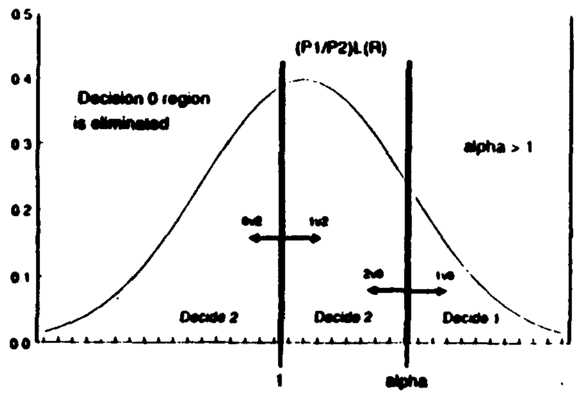


Figure 13. Case 2, $\alpha > 1$. Decision d_0 is completely eliminated.

From all the cases discussed above, it is apparent that if the decision rule is chosen to minimize a certain decision cost, then the indecision region depends on the choice of the associated costs. Hence, the probability masses can be assigned to the different propositions (decisions) in an optimal fashion so that the total risk is minimized, instead of being assigned arbitrarily as in the D-S theory.

Combining Rule

Let $U =: \{u_1, u_2, \ldots, u_N\}$ be the set of peripheral sensor decisions at the fusion center. Each u_i belongs to the set $\{d_0, d_1, d_2\}$. Let w_{ij} be the cast associ-

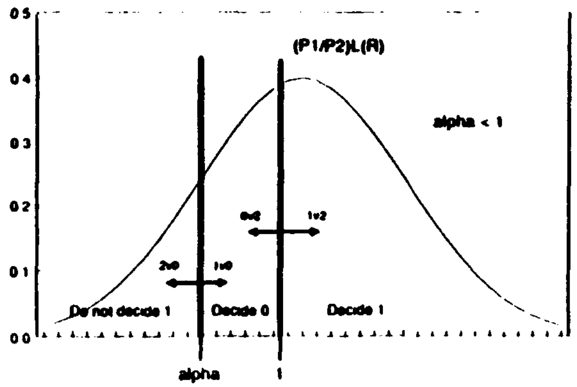


Figure 14. Case 2, $\alpha < 1$. Creation of a fuzzy decision region. "Do not decide d_1."

ated with the fusion deciding in favor of proposition d_i when the true hypothesis is H_j. If u designates the decision of the fusion, the total cost at the fusion is then

$$\mathbf{R}_f = \sum_i \sum_j w_{ij} P_j \int_{I_i} dP(u|H_j) \tag{15}$$

Assuming that the decisions from the peripheral sensors are independent conditioned on each hypothesis, (15) can be written as

$$\begin{aligned}
\mathbf{R}_f &= \sum_i \sum_j w_{ij} P_j \int_{I_i} \prod_{k=1}^{N} dP(u_k|H_j) \\
&= \sum_i \sum_j w_{ij} P_j \prod_{k=1}^{N} \left[\int_{I_i} dP(u_k|H_j) \right] \\
&= w_{00} P_0 \prod_k \left[\int_{I_0} dP(u_k|H_0) \right] + w_{10} P_0 \prod_k \left[\int_{I_1} dP(u_k|H_0) \right] \\
&\quad + w_{20} P_0 \prod_k \left[\int_{I_2} dP(u_k|H_0) \right] + w_{01} P_1 \prod_k \left[\int_{I_0} dP(u_k|H_1) \right] \\
&\quad + w_{11} P_1 \prod_k \left[\int_{I_1} dP(u_k|H_1) \right] + w_{21} P_1 \prod_k \left[\int_{I_2} dP(u_k|H_1) \right]
\end{aligned} \tag{16}$$

$$\begin{aligned}
&= \int_{I_0} [w_{00} P_0 dP(u|H_0) + w_{01} P_1 dP(u|H_1)] \\
&\quad + \int_{I_1} [w_{10} P_0 dP(u|H_0) + w_{11} P_1 dP(u|H_1)] \\
&\quad + \int_{I_2} [w_{20} P_0 dP(u|H_0) + w_{21} P_1 dP(u|H_1)]
\end{aligned} \tag{17}$$

The decision rule that minimizes the total risk assigns a particular combination of peripheral decisions U to that region that gives rise to the smallest integrand. Assuming that $w_{ii} = 0$, i.e., that there is no penalty for deciding correctly (a reasonable assumption in evidence processing), and that $w_{ij} - w_{2j} > 0$ for every j, i.e., that the cost of indecision is lower than the cost of deciding incorrectly, the test at the fusion becomes

$$\Lambda(u) \underset{d_0 \text{ or } d_2}{\overset{d_1 \text{ or } d_2}{\gtrless}} \frac{P_0 w_{10}}{P_1 w_{01}} \tag{18}$$

Similarly,

$$\Lambda(u) \underset{d_0 \text{ or } d_1}{\overset{d_2 \text{ or } d_1}{\gtrless}} \frac{P_0}{P_1} \frac{w_{20}}{w_{01} - w_{21}} \tag{19}$$

and

$$\Lambda(u) \underset{d_2 \text{ or } d_0}{\overset{d_1 \text{ or } d_0}{\gtrless}} \frac{P_0}{P_1} \frac{w_{10} - w_{20}}{w_{21}} \tag{20}$$

where

$$\Lambda(u) = \frac{dP(u|H_1)}{dP(u|H_0)} = \frac{dP(u_1|H_1)\ldots dP(u_N|H_1)}{dP(u_1|H_0)\ldots dP(u_N|H_0)}$$

$$= \prod_{i \in S_1} \frac{P_{D_i}}{P_{F_i}} \prod_{j \in S_0} \frac{1 - P_{D_j} - P_{I^1_j}}{1 - P_{F_j} - P_{I^0_j}} \prod_{n \in S_2} \frac{P_{I^1_n}}{P_{I^0_n}} \tag{21}$$

where $P_{I^1_j}$, $P_{I^0_j}$ indicate the probability masses at sensor j associated with the fuzzy decision (or, indecision) d_{uv_i} under the hypotheses H_1 and H_0 respectively. S_1 is the set of those decisions from the set U which favor $d_1(=H_1)$. S_0 is the set that favors $d_0(=H_0)$, and S_2 is the set from the peripheral decisions U that favors $d_2(=\{H_0 \text{ or } H_1\}$ – i.e., the undecided). Naturally, $U = S_0 + S_1 + S_2$.

From (18), (19), and (20), it follows that the optimal decision rule at the fusion is a likelihood ratio test. If we look at Eq. (21), the distribution of the LR under the two hypotheses is given by

$$P(\log\Lambda(u)|H_i) = P(\log\Lambda(u_1)|H_i) * \ldots * P(\log\Lambda(u_N)|H_i);\ i = 0, 1 \tag{22}$$

where "$*$" indicates convolution, with

$$P(\log\Lambda(u_i)|H_0) = (1 - P_{F_i} - P_{I^0_i})\delta\left(\log\Lambda(u_i) - \log\frac{1 - P_{D_i} - P_{I^1_i}}{1 - P_{F_i} - P_{I^0_i}}\right)$$

$$+ P_{I^0_i}\delta\left(\log\Lambda(u_i) - \log\frac{P_{I^1_i}}{P_{I^0_i}}\right) + P_{F_i}\delta\left(\log\Lambda(u_i) - \log\frac{P_{D_i}}{P_{F_i}}\right) \tag{23}$$

and $\delta(x)$ is the Kronecker's delta function, i.e. $\delta(x) = 1$ if $x > 0$ and zero otherwise, and

$$P(\log\Lambda(u_i)|H_1) = (1 - P_{D_i} - P_{I^1_i})\delta\left(\log\Lambda(u_i) - \log\frac{1 - P_{D_i} - P_{I^1_i}}{1 - P_{F_i} - P_{I^0_i}}\right)$$

$$+ P_{I_i}\delta\left(\log\Lambda(u_i) - \log\frac{P_{I^1_i}}{P_{I^0_i}}\right) + P_{D_i}\delta\left(\log\Lambda(u_i) - \log\frac{P_{D_i}}{P_{F_i}}\right) \tag{24}$$

Hence, the distribution of $\Lambda(U)$ under H_0 is given as the product of all possible combinations of $[(1 - P_{F_i} - P_{I^0_i}), P_{I^0_i}, P_{F_i}]$ according to their abscissae.[1] Similarly, the distribution of $\Lambda(U)$ under H_1 is given as the product of all possible combinations of $[(1 - P_{D_i} - P_{I^1_i}), P_{I^1_i}, P_{D_i}]$ according to their abscissae. Then, the fusion is done by using the appropriate thresholds from (18), (19) and (20). In that case, the probability masses (or beliefs) associated with each decision are combined according to the threshold and their abscissae. Thus, the combining rule involves pairwise multiplication of probability masses according to Table V, as in D-S theory, but the mass association is not done arbitrarily by using the mass that corresponds to conflict to renormalize the probability masses as in the D-S theory. In the GEP theory, the masses are associated via thresholds in an optimal way so that a certain risk (Eq. (17)) is minimized, or so that the probability of detection is maximized for fixed false alarm and indecision probabilities (generalized Neyman-Pearson test).

Table V. Generalized theory evidence combing rule.

	S2		
S1	$m_i^2(d_0)$	$m_i^2(d_1)$	$m_i^2(d_2)$
$m_i^1(d_0)$	$m_i^1(d_0)m_i^2(d_0)$	$m_i^1(d_0)m_i^2(d_1)$	$m_i^1(d_1)m_i^2(d_2)$
$m_i^1(d_1)$	$m_i^1(d_1)m_i^2(d_0)$	$m_i^1(d_1)m_i^2(d_1)$	$m_i^1(d_1)m_i^2(d_2)$
$m_i^1(d_2)$	$m_i^1(d_2)m_i^2(d_0)$	$m_i^1(d_2)m_i^2(d_1)$	$m_i^1(d_2)m_i^2(d_2)$

where the probabilities in Table V are conditioned on each hypothesis, and $i = 0, 1$. Thus, each m_i^j, $j = 1, 2$, in Table V is a conditional probability for $i = 0, 1$. Hence, the initial probability combining takes place among conditional probabilities only. For $i = 0, 1$, each product term in Table V, is a probability mass on the LRT coordinate axis with abscissa $m_i^1(d)/m_i^0(d)$ for every $d = d_0, d_1, d_2$. Evidence combining under each hypothesis is done from Table V by summing the probabilities from Table V whose abscissae fall in specific intervals specified either by an optimization criterion, or a certain desired performance. Hence, for $d = d_0, d_1, d_2$, evidence combining under each hypothesis H_i, $i = 0, 1$, is done according to the threshold rule

$$m_i^1(d_k)m_i^2(d_m) \to \text{decision } d_j \text{ if } \frac{m_1^1(d_k)m_1^2(d_m)}{m_0^1(d_k)m_0^2(d_m)} \in F_j \tag{25}$$

i.e.,

$$m_i^1(d_k)m_i^2(d_m) \to \text{decision } d_j \text{ if } t_j < \frac{m_1^1(d_k)m_1^2(d_m)}{m_0^1(d_k)m_0^2(d_m)} < t_{j+1} \tag{26}$$

for all k, m, and j, where t_j are the thresholds of the LRT's associated with the different decisions that minimize some risk function.

If multiple hypotheses (more than two) are tested, the combining rule remains the same as far as combining the belief function of the individual sources at the fusion to generate the new conditional belief function under each hypothesis. However, the association of the new belief function at the fusion with the set of admissible decisions must be done by using the multiple hypotheses LRT,[12] or another test that optimizes some performance measure.

Thus, evidence combining at the fusion is done conditioned on each hypothesis separately. The evidence is then associated with the admissible decisions unconditionally using an LRT or a test that optimizes some performance measure. Notice that the set of decisions need not be the same as the set of hypotheses. Thus, evidence combining and decision making are understood as separate concepts in the framework of the Generalized Evidence Combining Theory.

The generalization of the Bayesian (and N–P) theory by the Combined Theory is straightforward. An interpretation is probably required to incorporate the D-S theory into the Combined theory. If the probabilities $P(u_k = i|H_j)$, $i = 1, 2, 3$, are considered as the (conditional) bpa's (basic probability assignments) of the D-S theory for the kth sensor, $k = 1, 2, \ldots, \nu$, under hypothesis H_j, $j = 0, 1$, the evidence from the different sensors at the fusion is combined using the conditional distribution of the LR under the different hypothesis according to Table V. Hence, at the fusion a new (conditional) belief function is generated using the decision thresholds at the fusion. The (hard) decisions at the sensors are used to simply produce a hard decision at the fusion, if needed, according to some optimality criteria. In that respect, the GEP theory not only defines and processes the evidence according to an a priori set of optimality criteria, but also provides for hard decisions. It does so both at the sensors and fusion level according to the a priori set optimality criteria, which the D-S theory is incapable of doing, at least on a non-heuristic basis. (For example, in the D-S theory a hard decision can be made if the belief function exceeds a certain threshold. However, an increase of the belief in a given proposition does not necessarily imply that the belief in a contradictory propositions decreases. If this is the case, it is very difficult to make hard decisions with the D-S theory, even if needed.)

In the D-S theory the combining rule circumvents the problem of conflicting propositions by neglecting their masses and renormalizing the masses of the nonconflicting propositions. The renormalization is done by dividing the masses of nonconflicting propositions by the mass associated with the conflicting propositions. The combined support (belief) function is defined by

$$m = m_1 \oplus m_2 = \frac{\sum_{A_i B_j} m_1(A_i) m_2(B_j)}{1 - \sum_{A_k B_m = \emptyset} m_1(A_k) m_2(B_m)} \qquad (27)$$

where m_1 and m_2 designate the support (belief) functions from the two different sources of evidence defined over the same frame of discernment, and "$\emptyset$" is the empty set.[7] However, in the theory that we introduced there are no conflicting masses if the costs w_{ij} are properly chosen to avoid conflict. Furthermore, the combined evidence processing theory can be used to choose the proper

thresholds in the combining rule either to minimize the total risk or maximize the detection probability for fixed false alarm and indecision probability. Both these properties are highly desirable in sensor integration and decision fusion problems where we want to design a fusion system so that its performance can be assessed a priori and according to certain design criteria and specifications.

Hence, the new theory that we introduce, combines the advantages of the Bayesian theory, as far as optimizing the performance of the decision (fusion) system is concerned according to a priori determined optimality criteria, with the flexibility of the D-S theory, by allowing fuzzy decisions to be considered. Furthermore, it avoids the weaknesses of the D-S theory related to the choice of bpa's and the conflicting evidence in the Dempster's combining rule. Moreover, when needed, it allows for hard decisions at both sensors and fusion level. The D-S theory is incapable of doing so, at least in an unambiguous way.

DATA FUSION AT THE LEVEL OF DYNAMICS

At the level of dynamics, a mathematical model of the phenomenon or the process from which data is being collected using different sensors is required. The mathematical model need not be identical for each sensor. For example one sensor may only monitor a subset of the parameters that describe the phenomenon, or the states that describe the process. Yet, a global fusion model that integrates all the submodels that are used at the sensors must exist for coherent data fusion.

Since the observations at the sensors constitute a transformation of the parameters of the phenomenon or the states of the process into the observation space, a coherent observation model can be built at the fusion to integrate the observations from the different sensors.

Let $\mathbf{x}$ be the set of parameters or states that describe the phenomenon or the process under observation by sensor i in some convenient coordinate system $\mathbf{C}_i$. We assume then that there is a mathematical model in the form of

$$\mathbf{g}_i(\mathbf{x}_i, \mathbf{x}_i^{(k)}, k = 1, \ldots, K_i; u_{i,m}, m = 1, \ldots, M_i; w_{i,n}, n = 1, \ldots, N_i; t) = 0 \qquad (28)$$

that describes the phenomenon or the process from which data is collected, where $\mathbf{x}^{(k)}$ designates the kth order derivative, $u_{i,m}$ are possible known inputs (controls), and $w_{i,n}$ are unknown disturbances (deterministic or stochastic), and t designates time. Associated with the dynamic model (28) there is an observation model

$$\mathbf{z}_i = \mathbf{f}_i(\mathbf{x}_i) \qquad (29)$$

where $\mathbf{z}_i$ indicates the measurements at the ith sensor and $\mathbf{f}_i$ is the transformation from the parameter or state space $\mathbf{X}_i$ into the observation space $\mathbf{Z}_i$, i.e.,

$$\mathbf{f}_i: \mathbf{X}_i \to \mathbf{Z}_i \tag{30}$$

The transformation $\mathbf{f}_i$ can be either deterministic or stochastic.

Depending on the scenario, a sensor may either pass its observations to the fusion directly as raw data, or it may process it into some form of "meaningful" information. By the word meaningful we imply that the information from the sensor has a utilitarian value at the fusion. *Let the meaningful information from* the i-th sensor that process its data be designated by $\mathbf{z}_i$. The information $\hat{\mathbf{z}}_i := \hat{\mathbf{z}}_i(\mathbf{z}_i)$ can be a qualitative estimate of the parameters of the phenomenon or the process states, or some form of evidence that may support a certain proposition that can be used to *determine the true nature of a hypothesis that affects* the parameters or the state of the phenomenon or the process that is observed.

Hence, unprocessed (raw) and processed data from the different sensors may be present at the fusion. Before this data is fused, it should be converted into a common coordinate frame. The transformation from the sensor coordinate frame into the common one may be nonlinear which can complicate any further processing and fusion. If the translation of all data from the different sensors into a common framework is not forthcoming, it may be preferable to obtain estimates from some of the sensors first, and subsequently fuse the estimates with the remaining data.

Let $\mathbf{R}$ designate the subset of the sensors whose data is converted into a common coordinate frame and fused together at the fusion. Also, let $\mathbf{P}$ designate the subset of sensors that either transmit processed data to the fusion or whose data have been processed separately at the fusion before being converted into *the common coordinates system. Then, the fusion combines the data according* to some rule

$$\hat{\mathbf{z}} := \hat{\mathbf{z}}(\mathbf{z}_i, \hat{\mathbf{z}}_j(\mathbf{z}_j); i \in \mathbf{R}, j \in \mathbf{P}) \tag{31}$$

The rule $\hat{\mathbf{z}}$ is a mapping that fuses the information from the sensors into some desired form of semantically meaningful form. The form can be estimates of the parameters of the observed phenomenon or the process states, evidence supporting some proposition, a decision about the true nature of a hypothesis, or may constitute a reconstruction of the parameters or the states themselves as in image fusion for example.

From (31) it is seen that the fusion method is application driven, since the transformations $\hat{\mathbf{z}}(\mathbf{z}_i, \hat{\mathbf{z}}_j(\mathbf{z}_j); i \in \mathbf{R}, j \in \mathbf{P})$ and $\hat{\mathbf{z}}_j(\mathbf{z}_j), j \in \mathbf{P}$, must be chosen so that the objective at the fusion *is accomplished in the best possible way.*

A SENSOR INTEGRATION AND DATA FUSION PARADIGM

To illustrate the theoretical concepts that were developed in the previous sections, we consider a sensor integration and data fusion paradigm that incorporates all three different levels of data fusion described in the architecture of Figure 1. The paradigm involved a multiple sensor system that can be used for object recognition, identification, and tracking. Similar problems arise in autonomous vehicles, intelligent robots, *target trackers, space navigators*, etc.

We consider that data from an object sustaining translational and rotational motion in the three dimensional space is collected using a pulse radar,[12] a range radar[12,33] and a stereo camera.[27,31,34] The data from the different, and in particular, diversified sensors need to be integrated together. We proceed to describe one possible sensor integration approach by analyzing the necessary stages required to track the moving object.

Since we would like the fusion to exhibit "intelligent-like" behavior, we must assume that it must be capable of first recognizing the existence of an object *before tracking it*. Hence, the data should be fused so that the presence of an object can be detected and its identity identified first. Once this is done, estimates of the position, velocity, and possibly acceleration, must be generated by fusing the proper data together in order to track the object.

Hence, the first stage of data fusion should involve the object detection and identification. This can be done separately from the tracking, or it can be integrated with it depending on the data that is used for detection and identification and on the fusion architecture. If stereo images are used for detection and identification, data registration* between the left and right images of the stereo camera is required first. The *registered images can be used to reconstruct* the object for detection and identification purposes, recover depth, or obtain estimates of the object position, velocity, or acceleration depending on the fusion architecture. For image registration the artificial neural network described in Figures 2 and 3 can be used. Once the object stereo image is reconstructed by *standard image processing or artificial neural network techniques*, the existence of the object can be detected using either correlation and matching techniques or content addressable methods. Similar techniques can be used to identify and classify the object.

So far in the described processing the sensor integration involved only the registration of pixels in the stereo imagery. The detection and identification was then done using the stereo data exclusively. However, if data from a pulse radar or radars is also available as in the paradigm that is considered, we would like to take it into consideration, along with *data from the stereo images*, in order to enhance the detection and identification capabilities of the system. In the paradigm at hand, the integration of stereo image data with pulse radar(s) data constitutes the first stage of data fusion in the data sensor integration process that involves data from different types of sensors.

In this particular paradigm, where image data is fused with pulse radar data, the different format of the data from the camera and the radar does not allow for a direct integration of the information at the data level. Thus a transformation is required first before fusion takes place. The integration of the pulse radar(s) data with the stereo image data can take place at the evidential level, the second level in the architecture of Figure 1. Evidence provided by the pulse radar(s)

*Data registration is the association of pixels from the right image with pixels from the left image that correspond to the same object, or scene *in general* points.

and the stereo cameras can be combined using either the Generalized Evidence Processing theory, the Bayesian, or the Dempster-Shafer approach. No matter what the evidence combining approach is, a measure of confidence needs to be associated with the evidence provided by the cameras and the radar(s) about the presence or absence of the target. It is very important to note that in evidence combining the reliability of each evidence source must be integrated in the design of the fusion. Monotonicity of the detection and identification capabilities mast be enhanced and not be compromised after fusion.[1,18]

In the stereo camera(s) case, a degree of confidence can be associated with the decision based on the stereo images by taking into account the visibility conditions at the time the images were taken, the accuracy of the processing algorithm, and the resolution of the cameras. For the radar data, the detection and false alarm probabilities can be used as a measure of confidence on the radar decisions. Once the degrees of confidence, i.e., the measures of support, are available, the evidence from the cameras and the radar(s) can be fused into a final decision supporting either the presence of the object, or its absence. Depending on the permissible decisions, the result of the evidence fusion may be inconclusive too.

For a single radar the generic model that is used for target detection is

$$z = \begin{cases} s + n \text{ if } H_1 \text{ is true} \\ n \text{ if } H_0 \text{ is true} \end{cases} \tag{32}$$

where s is the known signal and n the noise. For the model (32), the presence of the target can be detected using the decision rules (9) through (11) in the GEP theory. The associated belief functions can be obtained from the partition of the observation space induced by the decision boundaries and the statistical assumptions about the noise n in the model (32). The evidence of the cameras and the radar(s) can be then combined into final inference regarding the presence of the object (target) using Table V for evidence combining and rule (24) for decision making.

Once the presence of the object is detected, or suspected in the case of inconclusive evidence, the available data from the sensors can be fused at the dynamics level to generate position, velocity, and acceleration estimates.

For fusion at the level of dynamics it is required that a mathematical model of the observed phenomenon or the process is available. For the tracking paradigm, it is assumed that the moving object is either maneuvering with constant velocity or constant acceleration. In both cases, the moving object model can be described by a linear stochastic differential equation driven by noise

$$\dot{x}_i(t) = A_i x_i(t) + G_i w(t) \tag{33}$$

where $x_i(t)$ is the state of the moving object, A_i the state matrix, G_i a known matrix, and $w(t)$ the noise process. The noise process can be modeled according

to the information available about the moving object. Usually, it is assumed that $w(t)$ is zero mean, white, gaussian noise. The choice of the state vector in the model (32) may also depend on the choice of the sensor(s). For example, in the camera case the natural choice is the standard Cartesian coordinates since they naturally give rise to the perspective projection measurements describing the transformation from the 3-D object space onto the 2-D camera(s) image plane. In the case of range radar, it is more convenient to use the polar coordinates instead of the Cartesian ones since they are more noise immune.[32] If different coordinate systems are used for different sensors, a common coordinate system must be chosen and all the data must be transformed to this common system before fusion. For the sake of simplicity, we assume that the measurements from both the camera(s) and the range radar(s) are converted to a Cartesian coordinate system, i.e., that both camera(s) and range radar(s) use the same state definition in the dynamical model (33).

For each sensor (camera or range radar in our paradigm) an observation equation is associated with the process Eq. (33). The observation model is a transformation, linear or nonlinear, from the state space onto the measurement space and has the general form

$$z_i(t) = H_i(x_i(t)) + v_i(t) \tag{34}$$

where the transformation $H_i(\ )$ can be linear or nonlinear and depends on the attributes of the moving object that are being measured and on the sensors physics. The term $v_i(t)$ is the observation process noise which is usually assumed to be a zero mean, white, Gaussian process uncorrelated or correlated with the process noise $w(t)$. Different noise models can be used depending on the particular sensor. For a more detailed account on this subject the reader is referred to Refs. 26 and 30.

In the paradigm at hand, let $z_{c,l}(t)$ and $z_{c,r}(t)$ indicate the position measurements of the state vector of a point of the object after registration. The corresponding transformations $H_{c,l}$ and $H_{c,r}$ are then nonlinear and are given by the projection equations.[26] For example, the transformation between (x, y, z) points of the object and pixel measurements on the left and right images are

$$x_{c,l} = f(x - d)/(f - z) \tag{35}$$

$$x_{c,r} = f(x + d)/(f - z) \tag{36}$$

in the x-axis direction, and

$$y_{c,l} = y_{c,r} = fy/(f - z) \tag{37}$$

in the y-axis direction, assuming that the image planes of the two cameras are on the same plane parallel to the xy-plane of the coordinate system. The transformations (35)–(37) are nonlinear. Similar transformations can be obtained for the velocities in the x and y-axis if the optical flow is measured and used at the fusion.[26,31]

On the other hand, the measurement model for the range radar is linear in which case H_r in the observation model becomes the constant matrix H_{rr} = [1 0 0] if polar coordinates are used. If Cartesian coordinates are used the matrix H_{rr} measures the position of the object in the x, y, and z directions. The integrated measurement model that is used to fuse the stereo images along with data from the range radar becomes

$$\begin{vmatrix} z_c(t) \\ z_{rr}(t) \end{vmatrix} = \begin{vmatrix} H_c(x_r(t)) \\ H_{rr}x_r(t) \end{vmatrix} + \begin{vmatrix} v_c(t) \\ v_{rr}(t) \end{vmatrix} \quad (38)$$

or, in a compact form

$$z(t) = H(x(t)) + v(t) \quad (39)$$

where the matrix function H and the noises $v_c(t)$ and $v_{rr}(t)$ are easily identifiable from the equations (35)–(37). With the fused data observed model (38), tracking, velocity, and acceleration estimates can be obtained using the Extended Kalman Filter (Fig. 15).

If a polar coordinate system is used to model the moving object dynamics at the range radar, the state definition between the camera(s) and the range radar(s) is different. Hence, a transformation from the polar coordinate system to the Cartesian system (or, vice versa) as required before the data from the camera(s) and the range radar(s) can be fused together.

However, the transformation from the polar to Cartesian coordinate system is non linear[32] resulting in a highly undesirable nonlinear data fusion model,

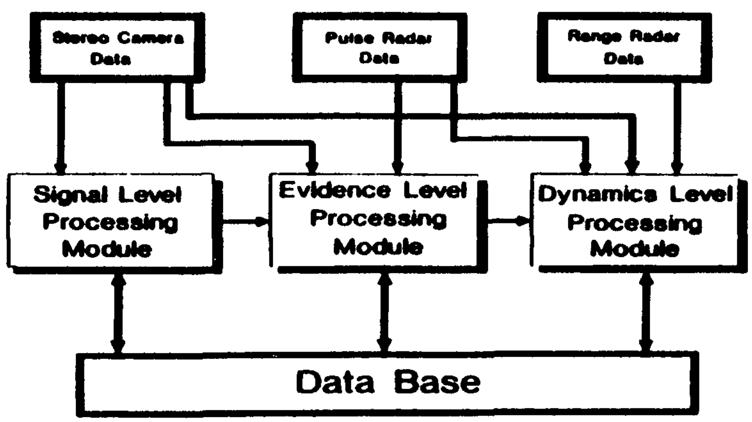


Figure 15. Data fusion for the object tracking paradigm

i.e., the fusion matrix function $H_c(t)$ becomes highly nonlinear. Hence, in view of these nonlinearities, it might be beneficial to group sensors that use the same or linearly related coordinate systems together and fuse their data separately from other troups of sensors that use different coordinate systems that are not linearly nrelated to the former. In the paradigm at hand, it might be preferable to fuse the data from the camera(s) and the range radar(s) separately to produce individual estimates in the respective coordinate systems, convert the estimates to a common coordinate system, and then fuse them together.

If $\hat{x}_{r,c}(t)$ and $\hat{x}_{r,rr}(t)$ indicate the estimates obtained by fusing the data from the camera(s) and the range radar(s) separately, the estimates can be fused, for example, according to

$$\hat{x}_r(t) = \alpha(t)\hat{x}_{r,c}(t) + \beta(t)\mathrm{T}[\hat{x}_{r,rr}(t)] \quad (40)$$

where T[.] designates the transformation from the Cartesian to the polar coordinate system, and $\alpha(t)$ and $\beta(t)$ are appropriate weighting factors that take into consideration the accuracy of the estimates, i.e., the associated covariance matrices. A possible form for these weighting factors might be:

$$\alpha(t) = [P_c(t)^{-1} + P_{rr}(t)^{-1}]^{-1}P_c(t)^{-1} \quad (41)$$

and

$$\beta(t) = [P_c(t)^{-1} + P_{rr}(t)^{-1}]^{-1}P_{rr}(t)^{-1} \quad (42)$$

where $P_c(t)$ and $P_{rr}(t)$ are the error covariances of the filters used for the camera(s) and range radar(s) data respectively.

The outlined fusion approach was used in Refs. 26, 35, and 36 to design a system for tracking objects sustaining 3-D translational and rotational motion by fusing data from stereo images and range radar measurements. Simulation of the fusion system has shown that fusion of the information as diversified as stereo images and range radar measurements can be fused together and result in a substantial improvement of the tracking accuracy even when the object sustains both translational and rotational 3-D motion. For a detailed description of the fusion system and extensive simulation results the reader is referred to Refs. 25, 35, and 36. An integrated model for multiple target tracking that fuses pulse radar data with range radar data asnd stereo images, and alleviates the need for separating the detection and identification part of the data fusion from the estimation (tracking) part is being developed by INTELNET.[30]

From the paradigm that is described in this section it should be clear that although a general fusion architecture that includes all three levels of fusion can be developed, there is no unique way for executing the functions of the three levels in a sequential or hierarchical fashion. Furthermore, a feedback from each one of the three levels to the other two may also be required to achieve the best utilization of theinformation from the different sensors. For example, in our paradigm, the estimates of the velocity and acceleration from the dynamic level module can be fed into a data base and be compared with the type of target that the object was identified with from the signal and evidence levels. The estimates can then be compared with other information that may exist in the data base to verify whether they can possibly correspond to the designated target. The outcome of the comparison can then be communicated to the three modules and fused back along with the sensors data in order to enhance the performance of the fusion.

CONCLUSION

An architecture for Sensor Integration and Data Fusion is presented. The architecture is based on three level processing modules. The three processing levels are the signal level, the evidential processing level, and the dynamical level. Each level requires different data representation and extracts different information from it. At the signal level an artificial neural network may be the most suitable processor when no mathematical description of the processed data is available. At the evidential level a Generalized Evidence Processing theory that has recently been introduced and combines the advantages of the Bayesian and the Demster-Shafer's theories without any of their drawbacks can be used. Finally, at the dynamical level a mathematical model that describes the phenomenon or the process from which data is gathered by the different sensors is required. The interconnections among the three level are dynamical and may be adapted to the particular fusion problem, the fusion objectives, and the physics of the sensors that are used. The fusion architecture includes a dynamic base as well. A paradigm of sensor integration and data fusion for object identificantion and tracking is presented. The paradigm refers to the fusion of data from a stereo camera(s), pulse radar(s), and range radar(s) and is used to illustrate and explain the steps that are involved in the fusion process when object identification and tracking is the objective of the fusion.

This research is sponsored in part by SDIO/IST and managed by the Office of Naval Research under Contract N00014-86-k-0515

REFERENCES

1. S. C. A. Thomopoulos, R. Viswanathan, and D. K. Bougoulias, "Optimal decision fusion in multiple sensor systems," *IEEE Transactions on Aerospace and Electronic Systems*, AES-23, 644–653 (1987).
2. R. Viswanathan, S. C. A. Thomopoulos, and R. Tumuluri, "Optimal serial distributed decision fusion," *IEEE Transactions on Aerospace and Electronic Systems*, AES-24, 366–376 (1988).
3. S. C. A. Thomopoulos, R. Viswanathan, and D. K. Bougoulias, "Optimal and suboptimal distributed decision fusion," *IEEE Transactions on Aerospace and Electronic Systems*, AES-25, 761–765, (1989)
4. S. C. A. Thomopoulos, D. K. Bougoulias, and L. Zhang, "Optimal and suboptimal distributed decision fusion," in *SPIE's 1988 Technical Symposium on Optics, Electro-Optics and Sensors*, 4–8 April 1988, Orlando, Florida
5. S. C. A. Thomopoulos, R. Viswanathan, and D. K. Bougoulias, "Optimal and suboptimal distributed decision fusion," in *22nd Annual Conference on Information Sciences and Systems*, Princeton University, March 16–18, 1988

6. S. C. A. Thomopoulos, R. Viswanathan, and D. P. Bougoulias, "Computable globally optimal distributed decision fusion," in *Conference on Decision and Control* (CDC '87), pp. 1846–1847
7. A. P. Dempster, "A generalization of Bayesian inference," *Journal of the Royal Statistical Society*, 30, 205–247 (1968)
8. G. A. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, 1976.
9. J. J. Chao, E. Drakopoulos, and C. C. Lee, "An evidential reasoning approach to distributed multiple-hypothesis detection," in *Proceedings of the 20th Conference on Decision and Control*, Los Angeles, CA, Dec. 1987, p. 1826–1831.
10. D. H. Krantz and J. Miyamoto, "Priors and likelihood ratios as priors," *Journal of the American Statistical Association*, 78 (1983), Theory and Methods 9
11. R. Viswanathan, A. Ansari, and S. C. A. Thomopoulos, "Optimal partitioning of observations in distributed detection," revised version submitted to *IEEE Transactions on Information Theory*.
12. H. L. Van Trees, *Detection, Estimation, and Modulation Theory*, Wiley, New York, 1968
13. Tenney, R. R. and Sandell, N. R., Jr., "Detection with distributed sensors," *IEEE Trans. on Aerospace and Electronic Systems*, AES-17, 501–510 (1981)
14. R. Srinivasan, "Distributed radar detection theory," *IEE Proceedings*, 133, 55–60 (1986)
15. Z. Chair and P. K. Varshney, "Optimal data fusion in multiple sensor detection systems," *IEEE Trans. on Aerospace and Electronic Systems*, AES-22, 98–101 (1986)
16. J. Tsitsiklis and M. Athans, "On the complexity of distributed decision problems," *IEEE Trans. on Automatic Control*, AC-30, 440–446 (1985)
17. S. C. A. Thomopoulos, R. Viswanathan, and D. K. Bougoulias, "Optimal decision fusion in multiple sensor systems," in *Proceedings of the 24th Allerton Conference*, October 1–3, 1986, Allerton House, Monticello, Illinois, pp. 984–993
18. S. C. A. Thomopoulos and N. Okello, "Distributed detection with consulting sensors and communication cost," in *22nd Annual Conference on Information Sciences and Systems*, Princeton University, March 16–18, 1988, pp. 891–895. Also in *SPIE's 1988 Technical Symposium on Optics Electro-Optics and Sensors*, Vol. 931, April 1988, Orlando, Florida, pp. 31–40.
19. S. C. A. Thomopoulos and N. Okello, "Decision fusion with consulting sensors: The Gaussian Case," in *American Control Conference ACC '88*, June 15–17, Atlanta, Georgia, pp. 419–424.
20. S. C. A. Thomopoulos and L. Zhang, "Distributed decision fusion with networking delays and channel errors," in *SPIE's 1988 Technical Symposium on Optics, Electro-Optics and Sensors*, Vol. 931, 4–8 April, 1988, Orlando, Florida, pp. 154–160
21. S. C. A. Thomopoulos and L. Zhang, "Distributed asynchronous Kalman filtering," in *SPIE's 1988 Technical Symposium on Optics, Electro-Optics and Sensors*, Vol. 931, 4–8 April, 1988, Orlando, Florida, pp. 161–170
22. S. C. A. Thomopoulos and L. Zhang, "Distributed filtering with random sampling and delay," in *Conference on Decision and Control*, CDC '88, Austin, Texas, Dec. 7–10, 1988, pp. 2348–53
23. S. C. A. Thomopoulos, "Distributed filtering and control in the presence of delays: Discrete and continuous case," submitted to *IEEE Trans. on Automatic Control*
24. S. C. A. Thomopoulos and A. D. Nestorides, "Distributed filtering and control in the presence of delays: Discrete time case," submitted to *Automatica*
25. S. C. A. Thomopoulos and N. N. Okello, "Distributed detection with mismatched sensors," in *SPIE 1988 Cambridge Symposium on Advances in Intelligent Robotics*

Systems, Boston, MA, Oct. '88. Also in the *Proceedings of CDC '88*, Austin, TX, Dec. '88, pp. 2495–2500.

26. L. H. Nilsson and S. C. A. Thomopoulos, *An Integrated Tracking System Based on Image Sequences*, Technical Report, Information Processing and Intelligent Systems (IPIS) Laboratory, Dept. of Electrical Engineering, Southern Illinois University, Carbondale, IL, April 1989.
27. B. K. P. Horn, *Robot Vision*, The MIT Press, Cambridge, MA, 1986.
28. D. Marr and T. Poggio, "Cooperative computation of stereo disparity," *Science* 194, 283–287 (1976).
29. S. C. A. Thomopoulos, "Generalized evidence processing theory," submitted to *IEEE Trans. on Aerospace and Electronic Systems*.
30. S. C. A. Thomopoulos, *Self-Organizing 3-D Vision System for Object Identification and Tracking*, Technical Report, INTELNET, 1988.
31. D. H. Ballard and O. A. Kimball, "Rigid body from depth and optical flow," *Computer Vision, Graphics, and Image Processing*, 22, 95–115 (1983).
32. S. S. Blackman, *Multiple-Target Tracking with Radar Applications*, Artech House, [illegible], Norwood, MA, 1986.
33. [illegible] DiFranco and W. L. Rubin, *Radar Detection*, Artech House, Inc., Dedham, [illegible], 1980.
34. L. H. Ballard and C. M. Brown, *Computer Vision*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1982.
35. S. C. A. Thomopoulos and L. Nillson, "Object tracking from image sequences using stereo camera and range radar," in *SPIE's Adv. in Intelligent Robotic Systems and Visual Communication and Image Processing '89 Conference*, Philadelphia, PA, Nov. 5–11, 1989.
36. L. Nillson, *An Integrated Tracking System Using a Stereo Camera and a Range Radar*, M.S. Thesis, Information Processing and Intelligent Systems (IPIS) Laboratory, Department of Electrical Engineering, Southern Illinois University, Carbondale, IL, June 1989.

ATION 363

othesis is H_j. If u

he fusion is

(III.13)

ors are independent

as

$\prod_{k=1}^{N} [\int_{F_i} dP(u_k | H_j)]$

(III.14)

$dP(u_k | H_i)]$

$\prod_k [\int_{F_i} dP(u_k | H_i)]$

$\prod_k [\int_{F_i} dP(u_k | H_i)]$

$\int_{F_i} [w_i P_i \, dP(u | H_i)]$

$] + w_i P_i \, dP(u | H_i)] [$

(III.15)

ial risk assigns a

to that region that

at $w_{ii} = 0$, i.e. that

nable assumption in

every j, i.e. that the

DECISION AND EVIDENCE FUSION IN SENSOR INTEGRATION

STELIOS C. A. THOMOPOULOS

Decision and Control Systems Laboratory
Department of Electrical and Computer Engineering
The Pennsylvania State University
University Park, PA 16802

I. INTRODUCTION

A. Distributed Decision Fusion and Evidence Processing

Sensor integration (or sensor fusion) was defined in [15] as "... the process of integrating raw and processed data into some form of meaningful inference that can be used intelligently to improve the performance of a system, measured in any convenient and quantifiable way, beyond the level that any one of the components of the system separately or any subset of the system components partially combined could achieve." A *taxonomy* for sensor fusion that involves three distinct levels at which

CONTROL AND DYNAMIC SYSTEMS, VOL. 49

information from different sensors can be integrated was proposed in [15]. According to the proposed taxonomy, sensor fusion can be accomplished at three different levels: the *signal* level, the level of *evidence*, and the level of *dynamics*. Each level is identified by the distinct features that the information that is fused carries, which (features) determine in turn the suitable processing methods for combining this information. The signal level in this taxonomy is characterized by the lack of a complete mathematical model that fits the data; the appropriate techniques for fusing information at this level are then correlation and learning through association. At the evidence level, a statistical model (not necessarily precise) describing the observed phenomenon or process is assumed; statistical techniques are then applied to fuse the information from different sensors. At the level of dynamics, a mathematical model that describes the process from which data are collected through some linear or nonlinear transformation of the process state. At this level, analytical tools can be used to fused the information from the sensors in either a centralized or decentralized way. The taxonomy in [15] is analogous to the three level - data, feature, and decision,- analysis that is used in classical pattern recognition problems. This chapter is primarily focused on different theories and approaches related to data fusion at the level of evidence. Data fusion at the evidence level will be referred to as decision or evidence fusion, depending on the semantic attributes associated with the fused information.

Distributed Decision (Evidence) Fusion (or DD(E)F in the sequel) exhibits some interesting characteristics which are not present in centralized, or raw data, fusion. The interesting characteristics relate to the semantic information that the decisions (in the broader sense of the term) convey which is not present, at least explicitly, when raw data

is fused. Different theories and results related to Distributed Decision Fusion (DDF) have appeared in the literature the last decade [TeSa 81, Sadj '86, ChVa '86, Srin '86, TVB '87, VTT '88, TVB '88, Demp '68, Shaf '76, Thom '90]. Each theory takes a different stand on the definition on how to measure evidence or combine decisions. The objective of this paper is to investigate the nature of DD(E)F, present some of the dominating theories on DDF and DEF, highlight similarities and differences among them that result from the semantic format of the fused information, and exploit natural topological equivalences between DDF and structures that exhibit learning abilities, such as neural networks.

To avoid concealing some of the issues under structural complexities and keep the discussion focused and as clear as possible we consider the simplest, yet fundamental, DDF topology and problem. We assume a parallel topology in which each sensor receives data from a common volume, Fig. 1. Furthermore, we assume that the sensors are perfectly aligned, so the problem of mismatch does not arise [13]. In this parallel topology we assume the simplest DDF problem with each sensor's data statistically independent from the other sensors. Each sensor performs a local operation on its data and transmits the outcome to the fusion. The fusion collects all the local information from the sensors and produces the global inference. In this framework, we consider both single-level logic decision rules, in which the number of permissible local decisions coincides with the number of hypotheses under test, and multi-level logic decision rules, in which the number of permissible local decisions exceeds the number of tested hypotheses. Most of the results in this paper pertain to the binary hypothesis testing problem. Extensions to multi-hypothesis testing are also included.

In DDF, the outcome of the global processing (fusion) depends on the outcome of the local data processing (sensor level) and the semantic format of the fused information. In the Bayesian context, the outcome of the local processing can be either hard decisions in a single-level logic, or soft decisions in a multi-level logic [16], or it can be the outcome of a simple quantization of the data, if no semantic attributes are attached to the outcome of the local processing [19]. In the context of the Dempster-Shafer's (D-S) theory, the outcome of the local processing is a set of probabilities that relate to the degree of support of the each proposition in the frame of discernment by the the data of each local processor [Demp '68, Shaf '76]. Thus, the local processing outcome of a Bayesian DDF is a quantized scalar number, whereas the outcome of the D-S local processor is a real-valued vector that corresponds to an entire probability distribution.

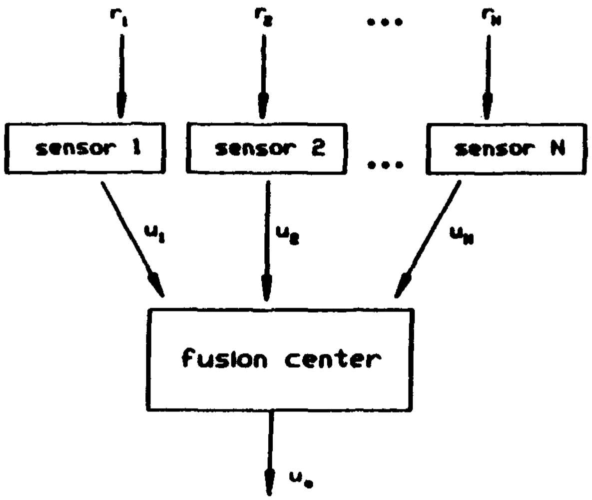


Fig. 1 Parallel sensor topology.

In addition to semantic differences in the output of the local processors, there are also substantial differences in the communication requirements for transmitting the local information to the fusion between the Bayesian DDF and the D-S DEF. Even in the presence of multi-level logic, the communication requirements for transmitting one out of, say, M integers is substantially lower than transmitting an M-dimensional real-valued vector. Hence, the communication requirements for the Bayesian DDF are substantially lower than the requirements of D-S DEF for the same number of data. Thus, a meaningful comparison between Bayesian and D-S DDF should either fix the available communication bandwidth to be the same for both approaches, or fix the fusion objectives to be common and study the communication overhead. In this paper we attempt a comparison of the D-S DDF with the Bayesian DDF assuming identical communication requirements.

The paper is structured as follows. The single-level logic, Bayesian DDF for the binary hypothesis testing problem is presented in Section 1. The theory is extended to multi-level logic and multiple hypotheses Bayesian DDF in Section 2. In Section 2 a generalized Bayesian framework for evidence processing is also presented. The D-S DDF theory is presented in Section 3 and theoretical as well as numerical comparative results between D-S DDF and D-S DDF are provided. The paper concludes with a discussion of topological similarities between the DDF structure and structures that exhibit learning capabilities. Numerical results that indicate the potential use of artificial neural networks in solving DDF problems are presented in Section 4.

B. Likelihood Ratio and Neyman-Pearson Test

In the multi-sensor detection related literature the common assumption that all the sensors cover the same geographical volume is made. Ref.'s [1] through [8]. Under this assumption, the mathematical model of binary hypothesis testing for each sensor becomes

$$z = \beta s + n \tag{I.1}$$

where s is the known signal, n is the noise, and β is a binary indicator random variable with $\beta = 1$ if H_1 is true and $\beta = 0$ if H_0 is true, thus, with probability distribution $p(\beta) = P_0\,\delta(\beta) + P_1\,\delta(\beta - 1)$. For the model (I.1) the optimal Bayesian detector is the Likelihood Ratio Test (LRT), namely

$$\Lambda(z) = \frac{p(z|H_1)}{p(z|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P_0\,(C_{10} - C_{00})}{P_1\,(C_{01} - C_{11})} := \frac{P_0}{P_1} T_C \tag{I.2}$$

where P_i, $i = 0, 1$, is the a-priori probability that hypothesis i is true with $P_1 = 1 - P_0$; C_{ij} is the cost associated with deciding in favor of hypothesis i while the true hypothesis is j, i = 0, 1, and j = 0, 1; and $T_C := \frac{(C_{10} - C_{00})}{(C_{01} - C_{11})}$, a constant factor that depends on the costs C_{ij} [9]. Alternatively, the LRT in (I.2) can be expressed as

$$p(z|H_1)P_1 \underset{H_0}{\overset{H_1}{\gtrless}} T_C\, p(z|H_0)P_0 \tag{I.3}$$

Noting that $p(z|\beta = i) = p(z|H_i)$, $i = 0, 1$, it follows easily from the model (I.1) that the LRT (I.2) or (I.3) is equivalent to the test

$$\beta^* = \frac{P_1\, p(z|\beta=1)}{p(z)} \underset{H_0}{\overset{H_1}{\gtrless}} T_1 \tag{I.4}$$

with threshold $T_1 = T_C P_0\; p(z|\beta=0) / p(z)$, where

$$\dot{\beta} = E[\beta|z] = P(\beta=1|z) = \frac{P_1\, p(z|\beta=1)}{p(z)} \tag{I.5}$$

is the minimum mean-squared error (mmse) estimate of β given the observation z.

In multi-sensor detection problems, the model (I.1), indexed by the I.D. number of each sensor, can be used to model the data of each sensor if all the sensors monitor the same, common geographical volume all the time. The data from the sensors can then be processed into a final decision either centrally using the *LRT* in the form of *(I.3)* or *(I.4)*, or locally first and then fused into a final decision by a fusion center.

II. Bayesian DDF in binary hypothesis testing with single-level local logic

Consider the binary hypothesis testing problem in a parallel sensor topology, Fig. 1. Assume that each sensor makes independent binary decisions and that the decisions are statistically independent from each other conditioned on each hypothesis. If u_i designates the i-th local decision, i.e. the decision of the i-th sensor, then

$$u_i = \begin{cases} +1 & \text{if the i-th local decison favors hypothesis } H_1 \\ -1 & \text{if the i-th local decison favors hypothesis } H_0 \end{cases} \tag{II.1}$$

Under these assumptions, the optimal Bayesian DDF rule that maximizes the detection probability for fixed false alarm probability at the fusion is given by the next theorem [14].

Theorem 1 [14] For the parallel sensor topology, binary hypothesis testing, single-level local logic, and statistically independent local decisions, the optimal Bayesian DDF that maximizes the fusion detection probability for fixed fusion false alarm probability consists of *Likelihood Ratio Tests (LRTs)* at the local (sensor) level and a *Neyman-Pearson (N-P) (possibly randomized) Test* at the fusion. □

A complete proof of the theorem can be found in [14]. Theorem 1 characterizes the optimal Bayesian DDF but does not provide with any means to determine the optimal local thresholds and the optimal fusion threshold. The problem of determining the optimal operating points in DDF is NP complete [5]. For the parallel topology, the number of possible solutions increases combinatorially with the number of sensors [TVB '88, '89] and, so far, no efficient algorithm exists for determining the optimal thresholds. A schematic representation of the optimal Bayesian DDF (to be referred to also as N-P (Neyman-Pearson) DDF rule when the false alarm at the fusion is fixed and we seek to minimize the detection probability) is shown in Fig. 2. As it will be discussed in Section 4, the structure of the optimal Bayesian DDF bares striking similarities with structures that exhibit learning capabilities, such as neural networks.

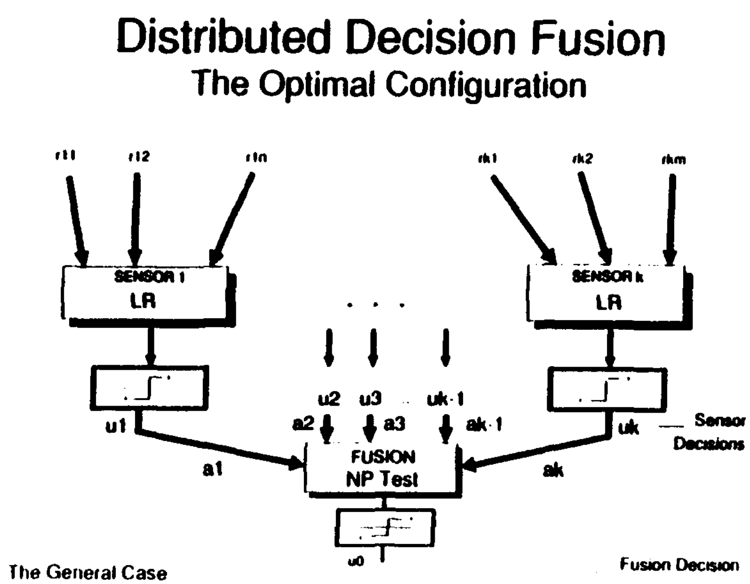


Fig. 2 Optimal Bayesian DDF.

Since it is computationally involved to determine the optimal thresholds for the Bayesian DDF, it is convenient from practical point-of-view to determine the optimal DDF when each local sensor operates at some fixed false alarm and detection probability. (In the sequel P_F will designate false alarm probability and P_D detection probability.) The optimal Bayesian DDF rule is then determined in Theorem 2 [7].

Theorem 2 [7] For the parallel sensor topology, the optimal Bayesian DDF rule when the local sensors operate at fixed false alarm and detection probabilities is the *LRT*

$$\Lambda(u) = \frac{dP(u \mid H_1)}{dP(u \mid H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} T_f \tag{II.2}$$

where $u =: (u_1, u_2, \ldots, u_N)$ is the vector of the peripheral decisions and T_f an appropriate threshold that is chosen so that a desired Bayesian risk function is optimized. If the sensor decisions are independent from each other conditioned on each hypothesis, the test in (3) simplifies to

$$\Lambda(u) = \frac{dP(u_1 \mid H_1) \ldots dP(u_N \mid H_1)}{dP(u_1 \mid H_0) \ldots dP(u_N \mid H_0)} = \prod_{i=1}^{N} \Lambda(u_i) \underset{H_0}{\overset{H_1}{\gtrless}} T_f \tag{II.3}$$

With the convention of (II.1), the Bayesian DDF takes on the form

$$\sum_{i=1}^{N} a_i u_i \underset{H_0}{\overset{H_1}{\gtrless}} T_f \tag{II.3a}$$

where

$$a_i = \begin{cases} \log\left(\dfrac{P_{Di}}{P_{Fi}}\right) & \text{if } u_i = +1 \\ -\log\left(\dfrac{1 - P_{Di}}{1 - P_{Fi}}\right) & \text{if } u_i = -1 \end{cases} \tag{II.3b}$$

where the index i indicates the i-th local processor (sensor). In a more compact form, the Bayesian DDF test (II.3) can be expressed as

$$\frac{1}{2}\sum_{i=1}^{N}\left((u_i+1)\log\left(\frac{P_{Di}}{P_{Fi}}\right)-(u_i-1)\log\left(\frac{1-P_{Di}}{1-P_{Fi}}\right)\right)\underset{H_0}{\overset{H_1}{\gtrless}} T_f \qquad \text{(II.3c)}$$

□

In the Bayesian context, determination of the optimal threshold generally requires: (a) knowledge of the a priori probabilities (likelihood) of the tested hypotheses, and (b) specification (subjective) of the attached costs C_{ij}, i.e. the cost of taking a decision j when the true hypothesis id i [VTr '68]. To eliminate these two requirements, the Neyman-Pearson (N-P) approach can be used to fuse the decisions [9]. The threshold T_f at the fusion is determined so that a desired false alarm probability is achieved. The following theorem summarizes the results.

Theorem 3 [9] For statistically independent sensors in parallel topology operating at fixed false alarm and detection probabilities, the optimal Bayesian DDF that maximizes the fusion detection probability for fixed false alarm probability is the *Neyman-Pearson (N-P) Test* (possibly *randomized*)

$$\Lambda(u)=\frac{dP(u_1\mid H_1)\ldots dP(u_N\mid H_1)}{dP(u_1\mid H_0)\ldots dP(u_N\mid H_0)}=\prod_{i=1}^{N}\Lambda(u_i)\underset{H_0}{\overset{H_1}{\gtrless}} T_f \qquad \text{(II.4)}$$

where N is the number of sensors. With the convention of (II.1), the N-P DDF takes on the form

$$\sum_{i=1}^{N} a_i u_i \underset{H_0}{\overset{H_1}{\gtrless}} T_f \qquad \text{(II.4a)}$$

where

$$a_i=\begin{cases}\log\left(\dfrac{P_{Di}}{P_{Fi}}\right) & \text{if } u_i=+1\\[2ex] -\log\left(\dfrac{1-P_{Di}}{1-P_{Fi}}\right) & \text{if } u_i=-1\end{cases} \qquad \text{(II.4b)}$$

The threshold T_f at the fusion is determined from the false alarm requirement by

$$\sum_{\Lambda(u)\geq T_f} dP(\Lambda(u)\mid H_0)=\alpha_0 \qquad \text{(II.4c)}$$

where α_0 is the desired false alarm at the fusion. In a more compact form, the Bayesian DDF test (II.3) can be expressed as

$$\frac{1}{2}\sum_{i=1}^{N}\left((u_i+1)\log\left(\frac{P_{Di}}{P_{Fi}}\right)-(u_i-1)\log\left(\frac{1-P_{Di}}{1-P_{Fi}}\right)\right)\underset{H_0}{\overset{H_1}{\gtrless}} T_f \qquad \text{(II.4d)}$$

For conditionally independent sensor decisions, the distribution of the LR at the fusion under the two hypotheses is given by

$$P(\log\Lambda(u)\mid H_i)=P(\log\Lambda(u_1\mid H_i)*\ldots*P(\log\Lambda(u_N)\mid H_i)\ ;\ i=0,1 \qquad \text{(II.5)}$$

where "*" indicates convolution with

$$P(\log\Lambda(u_i)\mid H_0)=(1-P_{Fi})\delta\left(\log\Lambda(u_i)-\log\frac{1-P_{Di}}{1-P_{Fi}}\right)+P_{Fi}\,\delta\left(\log\Lambda(u_i)-\log\frac{P_{Di}}{P_{Fi}}\right) \qquad \text{(II.6)}$$

and

$$P(\log\Lambda(u_i)\mid H_1)=(1-P_{Di})\delta\left(\log\Lambda(u_i)-\log\frac{1-P_{Di}}{1-P_{Fi}}\right)+P_{Di}\,\delta\left(\log\Lambda(u_i)-\log\frac{P_{Di}}{P_{Fi}}\right) \qquad \text{(II.7)}$$

Taking into account the discrete nature of the probabilities in (II.5), (II.6) and (II.7), the discrete distribution of the LR at the fusion consists of *non-zero probability points at abscissae of the form* $\prod_{i \in S} \frac{P_{Di}}{P_{Fi}} \prod_{j \in \bar{S}} \frac{1-P_{Dj}}{1-P_{Fj}}$ *where S is a subset of the sensors* {1, 2, ..., N} *that* favors hypothesis H_1 and $\bar{S}$ is its compliment that favors hypothesis H_0. The corresponding probabilities of the LR at these abscissae are $\prod_{i \in S} P_{Di} \prod_{j \in \bar{S}} (1-P_{Dj})$ *under* H_1 and $\prod_{i \in S} P_{Fi} \prod_{j \in \bar{S}} (1-P_{Fj})$ *under* H_0. □

Since convolution is both associative and commutative and the product terms of the discrete convolution for every two sensors in (II.5) are generated by cross-multiplying the conditional probabilities under each hypothesis, the N-P DDF combining rule (II.5, I.6, I.7) can be implemented using a table similar to the one used for the Dempster's combining rule in the D-S theory [Demp '68, Shaf '76; also Section 3]. By considering two sensors at a time, the conditional probabilities under *each hypothesis for each sensor are placed along the sides of the table*, and all possible combinations are formed by multiplying them pairwise (for illustration, see Table I in the Generalized Evidence Processing theory section). For binary hypothesis testing and hard decisions, the so created 2x2 tables are combined with each other, and the process is repeated until the convolution (II.5) is generated in the final table.

Once the final convolution table is obtained, association of the resulting probabilities with all possible events (decisions in this case) is done by sorting the entries in the table (convolution points) according to the numerical values of the corresponding abscissae that take the form $\prod_{i \in S} \frac{P_{Di}}{P_{Fi}} \prod_{j \in \bar{S}} \frac{1-P_{Dj}}{1-P_{Fj}}$. In the final stage, probability masses are allocated to the different alternatives (decisions) by choosing decision boundaries that optimize certain desired criteria at the fusion [TVB '87, '89]. In contrast with the D-S theory, *the rule* (II.5, I.6, I.7) *does not lead to inconsistent probability masses when the intersection of the corresponding events is empty as in D-S theory* [Demp '68, Shaf '76]. Thus, the need for (arbitrary) evidence renormalization required in D-S theory [Shaf '76] is avoided, and the possibility of simultaneous increase of evidence about conflicting proposition that may occur in D-S theory is prevented. Furthermore, in addition to evidence combining, the combining rule (II.5, I.6, I.7) provides decision regions and boundaries according to desired performance criteria chosen to optimize the performance of the fusion (and the peripheral sensors). *If the N-P DDF rule* (II.5, I.6, I.7) *is interpreted as an evidence combining rule with evidence the conditional probabilities* $dP(u_i|H_j)$, $i = 1, \ldots, N$ and $j = 0, 1$, some interesting comparisons are made with the Demspter's combining rule in Section 3.

Although Theorems 2 and 3 were independently proven in [ChVa'86] and [9] respectively and preceded the proof of Theorem 1, they can be directly derived from Theorem 1. In DDF with local processors operating at fixed false alarm and detection probability (P_F, P_D), it is legitimate to raise the question whether the performance after fusion improves beyond that of any sensor or subset of sensors in the configuration individually. The answer to this question turns out to be affirmative as proven in [9].

Theorem 4 [9] *In a configuration of N similar sensors, all operating at the same* $(P_F, P_D) = (p, q)$, *the randomized N-P test at the fusion can provide an overall* (P^f_F, P^f_D) *that satisfies*

$$P^f_F \le \min_{i \in N} \{P_{F_i}\} \text{ and } P^f_D > \max_{i \in N} \{P_{D_i}\}$$

provided that $N \geq 3$.

More precisely, for $N \geq 3$, the randomized N-P test can be fixed so that

$$P^f_F = P_F = p \text{ and } P^f_D > P_D = q$$

where P_F and P_D are the false alarm and detection probability at the individual sensors. □

Table I. Fusion system of five sensors in parallel topology. All sensors have the same false alarm probability $P_{f_i} = 0.05$, $i = 1, \ldots, 5$. The corresponding detection probabilities are as follows.*

i	1	2	3	4	5
P_{D_i}	0.95	0.94	0.93	0.92	0.91

Decision fusion: 5	Sensor system	
Sensors PF: Equal x	Unequal -	
Sensors PD: Equal -	Unequal x	

Threshold @ fusion center	Probability of detection @ fusion center	Probability of false alarm @ fusion center
PDMAX = 0.95000	PFMIN = 0.50000E-01	
t^*	PD	PF
6163.2	0.957817	0.300000E-04
53.004	0.963797	0.142812E-03
45.880	0.968973	0.255625E-03
40.339	0.973523	0.368437E-03
38.907	0.977913	0.481250E-03
34.208	0.981772	0.594062E-03
32.081	0.985391	0.706874E-03
29.610	0.988731	0.819687E-03
28.207	0.991913	0.932499E-03
24.416	0.994668	0.104531E-02
20.705	0.997003	0.115812E-02
0.20998	0.997454	0.330156E-02
0.17806	0.997835	0.544500E-02
0.15413	0.998165	0.758843E-02
0.14683	0.998480	0.973187E-02
0.13552	0.998771	0.118753E-01
0.12709	0.999043	0.140187E-01
0.11174	0.999282	0.161622E-01
0.10778	0.999513	0.183056E-01
0.94760E-01	0.999717	0.204490E-01
0.[illegible]2023E-01	0.999892	0.225925E-01

*After fusion, the detection probability is higher than the highest detection probability among all sensors for a wide range of false alarm probabilities lower than 0.05.

Theorem 4 raises two interesting points. First, that it is possible to create a "super-sensor" that meets strict performance requirements from inferior sensors that individually do not meet the performance requirements, thus legitimizing the idea of DDF. Second, it proves that nothing is gained in performance when binary decisions from only two identical sensors are fused; it takes at least three identical sensors to improve performance. Theorem 4 characterizes the DDF performance when all sensors are similar. Although no similar theoretical result exists for dissimilar sensors, extensive numerical simulations have shown that the conclusions of theorem extend to dissimilar sensors as well [TVB '87, '88]. Numerical results from the application of Theorem 4 in N-P DDF are summarized in Table I. Additional numerical results in the spirit of the theorem can be found in [9] and [16].

III. Bayesian DDF in binary hypothesis testing with multi-level local logic

For the binary hypothesis testing problem and similar assumptions as above, i.e. sensors in parallel topology and data statistically independent under either hypothesis from sensor to sensor, we consider the Bayesian DFF problem when the local processors (sensors) employ a multilevel logic. The problem is equivalent to utilizing a multi-level quantizer locally and a binary decision fusion rule. If the objective of optimally choosing the quantization levels is to maximize the detection probability at the fusion for fixed false alarm probability, the optimal quantizer is given by Theorem 5.

Theorem 5 [12] For the multi-level local logic Bayesian DDF, quantization of the local data according to *Likelihood Ratios (LTs)* maximizes the probability of detection for fixed false alarm probability at the fusion. Furthermore, the optimal fusion rule is the *N-P test* at the desired false alarm level. □

Proof The proof of Theorem 5, which is a generalization of the proof of Theorem 1, is based on the monotone property of the optimal Bayesian DDF rule [TVB '87, VAT '88] and the fact that the *LRT* satisfies this monotone property. A summary of the proof has appeared in [12]. The complete proof of the theorem is given in the Appendix. An alternative proof of the theorem has been recently given in [18].

Hence, in the Bayesian DDF the optimal fusion rule consists of LRTs at the local and global level. In the multi-level local logic DDF, the semantic correlation between quantization levels and decisions favoring one hypothesis or a group of hypotheses is not, in general, clear unless a specific decision is attached to each quantization level by minimizing some total decision cost. This is the idea behind the *Generalized Evidence Processing (GEP)* Theory [Thom '90, ThGa '90] which extends the single logic N-P (Bayesian) DDF to multi-level logic so that a correspondence between Bayesian decision processing and D-S evidence processing can be established.

A. Generalized Evidence Processing (GEP) Theory

The pivoting idea behind GEP theory is the separation of hypotheses from decisions. Once this separation is understood, the Bayesian (or N-P) DDF theory can be extended to a frame of discernment similar to that of D-S theory. In the context of GEP theory, the choice of different decisions can be thought off as different quantization levels of the data. For notational simplicity, the GEP theory is first presented for binary hypothesis decision fusion. Generalization to multiple hypotheses decision fusion follows at the end of the section.

Let H_0, H_1 be the two hypotheses under test. The probability space is partitioned into two regions according to the events $\{\omega = H_1\}$ and $\{\omega = H_0\}$ with associated probabilities $P_1 \geq 0$ and $P_0 \geq 0$ respectively, where $P_1 + P_0 = 1$.

Let d_0, d_1, and $d_2 := d_{0\vee 1}$ be a frame of discernment used by a decision maker to partition the probability space according to the gathered evidence, where the three decisions correspond to the propositions "H_0 true," "H_1 true," and "H_0 or H_1 true," respectively. The decision $d_2 := d_{0\vee 1}$, where "$\vee$" stands for "or," indicates the inability of the decision maker to come up with conclusive evidence on the true nature of the hypothesis.

In the classical probabilistic (Bayesian) framework, the probability associated with $d_{0\vee 1}$ is equal to

$$Pr[d_{0\vee 1}] = Pr[H_0 + H_1] = Pr[H_0] + Pr[H_1] = 1 \qquad \text{(III.1)}$$

since H_0 and H_1 constitute a disjoint coverage of the probability space over which the evidence processing problem is defined. As it was mentioned earlier, the apparent weakness of the Bayesian theory to incorporate non-mutually exclusive, i.e. redundant, propositions gave rise to the D-S theory which is particularly efficient in dealing with fuzzy propositions. However, by disassociating decisions from hypotheses, a unified framework is created which can accommodate both Bayesian and D-S DDFs.

Let z be the transformation from the initial event space Ω into the observation (data) space **Z**, i.e.

$$z: \Omega \rightarrow \mathbf{Z} \qquad \text{(III.2)}$$

and d be a transformation from the observation space into the decision space **D**, i.e.

$$d: \mathbb{Z} \rightarrow \mathbb{D} \tag{III.3}$$

Let $\{dP(z|H_i), P(H_i) : i = 0, 1\}$ be the probability measure on $\mathbb{Z}$. Let C_{ij} be the cost associated with a decision i when the true hypothesis is H_j. Define the mapping d so that the cumulative risk

$$\mathbb{R} = \sum_i \sum_j C_{ij} P_j \int_{Z_i} dP(z|H_j) \; ; j = 0, 1, \text{ and } i = 0, 1, 2 \tag{III.4}$$

is minimized, where $d_2 := d_{0 \vee 1}$ is the ambiguous decision. In (III.4), the regions Z_i indicate the partition of the observation space according to the decision rule d, where Z_i indicates the region of the observation space in which the decision is d_i. *Notice that the partition of the* observation space is made according to the set of decisions that is chosen a priori, and that in a given set a particular decision may not favor a specific hypothesis exclusively, in contrast with what is customary in the Bayesian theory. However, it is not clear from (III.4) if such a partition is feasible, since the decision rule is defined though R which depends on the partition $\{Z_i\}$. We show next that for a proper choice of the costs C_{ij}, there exist partitions $\{Z_i\}$ that define legitimate probability measures on $\mathbb{D}$, and thus a generalized decision rule d.

Rewriting equation (III.4) as

$$\mathbb{R} = \int_{Z_0} \{P_0 C_{00} dP(z|H_0) + P_1 C_{01} dP(z|H_1)\} + \int_{Z_1} \{P_0 C_{10} dP(z|H_0) + P_1 C_{11} dP(z|H_1)\} + \int_{Z_2} \{P_0 C_{20} dP(z|H_0) + P_1 C_{21} dP(z|H_1)\} \tag{III.5}$$

the total risk is minimized if the decision rule assigns z (the observation) to the region that corresponds to the least integrand under the three integrals in (III.5). Hence, the decision rule becomes

$$P_0 C_{00} dP(z|H_0) + P_1 C_{01} dP(z|H_1) \underset{Z_0 \text{ or } Z_2}{\overset{Z_1 \text{ or } Z_2}{\gtrless}} P_0 C_{10} dP(z|H_0) + P_1 C_{11} dP(z|H_1) \tag{III.6}$$

and symmetrically for the other alternatives. Dividing both sides of (III.6) by $dP(z|H_0)$ and defining $\Lambda(z) = \dfrac{dP(z|H_1)}{dP(z|H_0)}$, the decision rule becomes

$$(C_{01} - C_{11})\,\Lambda(z) \underset{d_0 \text{ or } d_2}{\overset{d_1 \text{ or } d_2}{\gtrless}} \frac{P_0}{P_1}(C_{10} - C_{00}) \tag{III.7}$$

Similarly,

$$(C_{01} - C_{21})\,\Lambda(z) \underset{d_0 \text{ or } d_1}{\overset{d_2 \text{ or } d_1}{\gtrless}} \frac{P_0}{P_1}(C_{20} - C_{00}) \tag{III.8}$$

and

$$(C_{11} - C_{21})\,\Lambda(z) \underset{d_1 \text{ or } d_0}{\overset{d_2 \text{ or } d_0}{\gtrless}} \frac{P_0}{P_1}(C_{20} - C_{10}) \tag{III.9}$$

From (III.7), (III.8), and (III.9), it is seen that the decision rule depends on *the relative values of the C_{ij} costs*. We examine three different cases to illustrate the significance of the C_{ij}'s.

Case 1 The associated cost for correct decision is zero, i.e. $C_{11} = C_{00} = 0$, while the cost of incorrect guessing is higher than the cost associated with indecision under both hypotheses, i.e. $C_{ij} > C_{2j}$ for every $i \neq 2$ and $j = 0$ or 1. Under these conditions, the decision rule becomes

TABLE I
Number Of Monotone Increasing Functions And Percentage of Reduction

Number of Sensors N	Number of Monotone Functions	Number of all Possible 2^{2^N} Functions	Percentage Reduction
1	3	4	25
2	6	16	62.5
3	20	256	92.19
4	168	65,536	99.74
5	7,581	4.2949673×10^9	99.99982
6	7,828,354	1.8446744×10^{19}	100

TABLE II
Total Number Of Functions Searched For The Set Of Optimal Thresholds

Number of Sensors N	L_N (is number of Monotone Functions -2)	Total Number of Functions R_N	Percentage Reduction
1	1	1	0.00
2	4	2	50.00
3	18	9	50.00
4	166	114	31.13
5	7,579	6,894	9.03
6	7,828,352	7,786,338	0.54

probability of detection at the fusion for the given false alarm probability. Let the best randomized N-P test at the fusion center be $t(u_1,\ldots,u_N) \gtrless^{H_1}_{H_0} \lambda_0$ w.p. p, resulting in false alarm probability P_{F_0}, and $\bar{t}(u_1,\ldots,u_N) \gtrless^{H_1}_{H_0} \bar{\lambda}_0$ w.p. $1-p$, resulting in false alarm probability $\bar{P}_{F_0}$. The thresholds λ_0 and $\bar{\lambda}_0$ are chosen so that the total false alarm at the fusion

$$P_{F_0} = pP_{F_0} + (1-p)\bar{P}_{F_0} = \alpha_0. \qquad (5)$$

Thus, the corresponding detection probability at the fusion

$$P_{D_0} = pP_{D_0} + (1-p)\bar{P}_{D_0}. \qquad (6)$$

Since the probability p is fixed from the constraint (5), the detection probability in (6) is maximized if each one of the P_{D_0} and $\bar{P}_{D_0}$ is maximized. But, according to the part of the proof in the nonrandomized N-P test above, each one of these two detection probabilities is maximized if an L-R test is used at the sensors. Hence, the Theorem is also proven for the randomized N-P/L-R test.

A precise characterization of the set of fusion functions that satisfy Theorem 1, indicated as R_N in Table II, can be found in [12].

III. CONCLUSIONS

A general proof that the optimal fusion rule for the distributed detection problem of Fig. 1 involves an N-P test (or a randomized N-P test) at the fusion and L-R tests at all sensors has been provided. The proof does not suffer from the weaknesses of the Lagrange-multipliers-based proof in [10].

S. C. A. THOMOPOULOS*
R. VISWANATHAN
D. K. BOUGOULIAS
Dep't. of Electrical Engineering
Southern Illinois University
Carbondale, IL 62901

*Currently with:
Dep't. of Electrical Engr.
The Pennsylvania State Univ.
University Park, PA 16802

REFERENCES

[1] Conte, E., D'Addio, E., Farina, A., and Longo, M. (1983)
Multistatic radar detection: synthesis and comparison of optimum and suboptimum receivers.
IEEE Proc. F, Commun., Radar & Signal Process. 1983.

[2] Tenney, R. R., and Sandell, N. R., Jr. (1981)
Detection with distributed sensors.
IEEE Transactions on Aerospace and Electronic Systems, AES-17 (July 1981), 501–510.

[3] Sadjadi, F. A. (1986)
Hypothesis testing in a distributed environment.
IEEE Transactions on Aerospace and Electronic Systems, AES-22 (Mar. 1986), 134–137.

[4] Teneketzis, D., and Varaiya, P. (1984)
The decentralized quickest detection problem.
IEEE Transactions on Automatic Control, AC-29, 7 (July 1984), 641–644.

[5] Teneketzis, D. (1982)
The decentralized Wald problem.
In *Proceedings of the IEEE 1982 International Large-Scale Systems Symposium*, Virginia Beach, VA, Oct. 1982, 423–430.

[6] Tsitsiklis, J., and Athans, M. (1985)
On the complexity of distributed decision problems.
IEEE Transactions on Automatic Control, AC-30, 5 (May 1985), 440–446.

[7] Chair, Z., and Varshney, P. K. (1986)
Optimal data fusion in multiple sensor detection systems.
IEEE Transactions on Aerospace and Electronic Systems, AES-22, 1 (Jan. 1986), 98–101.

[8] Thomopoulos, S. C. A., Viswanathan, R., and Bougoulias, D. K. (1986)
Optimal decision fusion in multiple sensor systems.
In *Proceedings of the 24th Allerton Conference*, Monticello, IL, Oct. 1–3, 1986, 984–993.

[9] Thomopoulos, S. C. A., Viswanathan, R., and Bougoulias, D. P. (1987)
Optimal decision fusion in multiple sensor systems.
IEEE Transactions on Aerospace and Electronic Systems, AES-23, 5 (Sept. 1987), 644–653.

[10] Srinivasan, R. (1986)
Distributed radar detection theory.
IEE Proceedings, 133, Pt. F, 1 (Feb. 1986), 55–60.

[11] Viswanathan, R., and Thomopoulos, S. C. A. (1987)
Distributed data fusion.
Technical report, TR-SIU-DEE-87-4, Department of Electrical Engineering, Southern Illinois University, Carbondale, Apr. 1987.

$$\Lambda(z) \underset{d_0 \text{ or } d_2}{\overset{d_1 \text{ or } d_2}{\gtrless}} \frac{P_0}{P_1}\frac{C_{20}}{C_{21}} \qquad \text{(III.10)}$$

Similarly,

$$\Lambda(z) \underset{d_0 \text{ or } d_1}{\overset{d_2 \text{ or } d_1}{\gtrless}} \frac{P_0}{P_1}\frac{C_{20}}{C_{01} - C_{21}} \qquad \text{(III.11)}$$

and

$$\Lambda(z) \underset{d_0 \text{ or } d_2}{\overset{d_1 \text{ or } d_0}{\gtrless}} \frac{P_0}{P_1}\frac{C_{10} - C_{20}}{C_{21}} \qquad \text{(III.12)}$$

In this case, it is seen from (III.10) through (III.12) that the optimal test (decision rule d) is of the likelihood type with the indecision region dependent on the relative values of C_{ij}. Different numerical applications are considered next to further illustrate the significance of the C_{ij}'s.

Case 1.1 $C_{00} = C_{11} = 0$, $C_{10} = C_{01} = 1$, and $C_{20} = C_{21} = 1/3$. In this case $\frac{C_{20}}{C_{21}} = 1$, $\frac{C_{20}}{C_{01} - C_{21}} = 0.5$, and $\frac{C_{10} - C_{20}}{C_{21}} = 2$. The partition of the LR by the decision rule in this case is shown in Fig. 3. From Fig. 3, *it is seen that in this case the indecision region lies between the two definite decision regions*, which corresponds to the way that we would intuitively have picked the indecision (uncertainty) region relative to a LRT [9].

Case 1.2 $C_{00} = C_{11} = 0$, $C_{10} = C_{01} = 1$, and $C_{20} = C_{21} = 0.5$. In this case $\frac{C_{20}}{C_{21}} = \frac{C_{20}}{C_{01} - C_{21}} = \frac{C_{10} - C_{20}}{C_{21}} = 1$, i.e. all three thresholds are the same, and the indecision region is completely eliminated. The partition of the LR by the decision rule is shown in Fig. 4. It corresponds to a standard binary hypothesis - binary decision Bayesian problem.

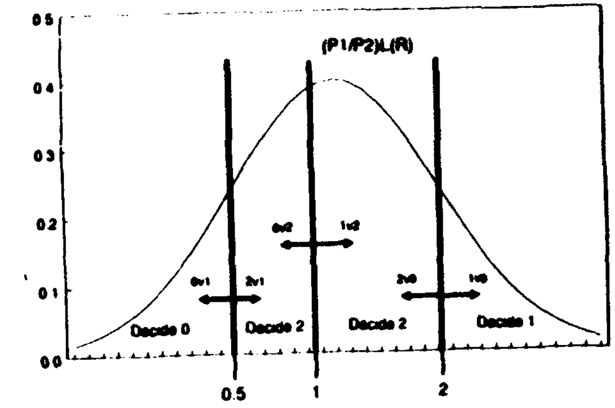


Fig. 3 Case 1.1 The indecision region lies between the two definite decision regions.

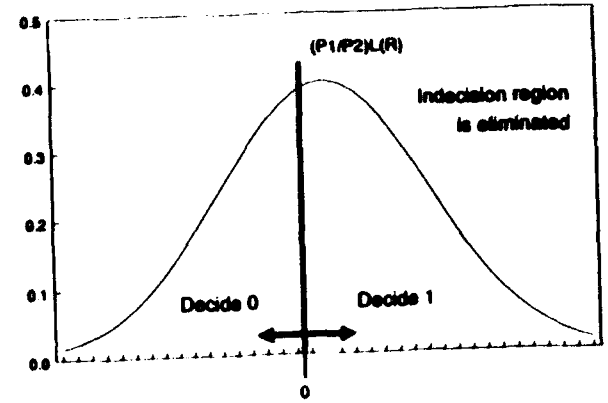


Fig. 4 Case 1.2 The indecision region is completely eliminated.

Case 1.3 $C_{00} = C_{11} = 0$, $C_{10} = C_{01} = 1$, and $C_{20} = C_{21} = 1/2$. In this case $\frac{C_{10}}{C_{01}} = 1$, $\frac{C_{20}}{C_{01} - C_{21}} = 2$, and $\frac{C_{10} - C_{20}}{C_{21}} = 1/2$. The partition of the LR by the decision rule in this case is shown in Fig. 5. In this case the two definite (hard) decision regions are sandwiched between the two indecision regions opposite to what one would intuitively have defined as an indecision region in an LRT, and exactly opposite to Case 1.1.

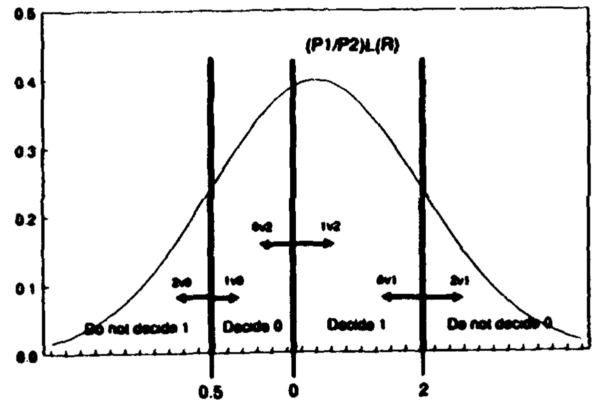


Fig. 5 Case 1.3 The definite decision regions lie between the indecision regions.

Case 2 $C_{00} = C_{11} = 0$, $C_{01} = C_{21}$. In this case $\frac{C_{10}}{C_{01}} = 1$, $\frac{C_{20}}{C_{01} - C_{21}} = \infty$, and $\frac{C_{10} - C_{20}}{C_{21}} = \alpha$. The partition of the LR into different decision regions is shown in Fig.s 6. and 7 depending on the

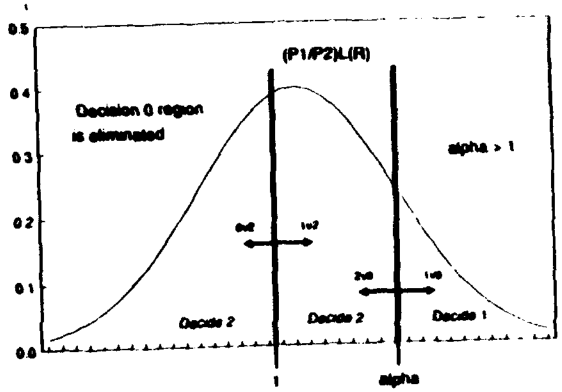


Fig. 6 Case 2, $\alpha > 1$. The decision d_0 is completely eliminated.

value of α. In Fig. 6, $\alpha > 1$, and the decision d_0 is completely eliminated. In Fig. 7, $\alpha < 1$ and there are three decision regions; d_1, d_2, and not d_1. According to the definition of the three possible decisions, the decision "do not decide d_1" must be interpreted as the fuzzy decision "d_0 or d_2" and not as "decide in favor of H_0."

Case 3 If in the above cases $C_{2j} > C_{ij}$, $i \neq 2$ and $j = 0, 1$, the LRT in (III.11) is reversed and the threshold in (III.12) becomes *negative*. *Under these circumstances*, the decision regions in the previous cases are reversed.

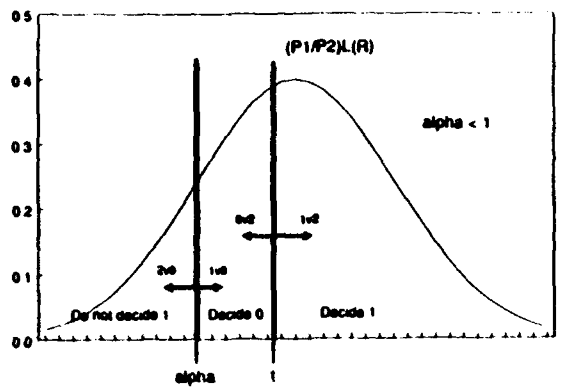


Fig. 7 Case $\alpha < 1$. Creation of a fuzzy decision region: "Do not decide d_1"

From all the cases discussed above, it is apparent that if the decision rule is chosen to minimize a certain decision cost, then the indecision region depends on the choice of the associated costs. Hence, the probability masses can be assigned to the different propositions (decisions) in an optimal fashion so that the total risk is minimized, instead of being assigned arbitrarily as in the D-S theory.

GEP Combining rule Let $u =: \{u_1, u_2, \ldots, u_N\}$ be the set of peripheral sensor decisions at the fusion center. Each u_i belongs to the set $\{d_0, d_1, d_2\}$. Let w_{ij} be the cost associated with the fusion

deciding in favor of proposition d_i when the true hypothesis is H_j. If u designates the decision of the fusion, the total cost at the fusion is

$$\mathcal{R}_f = \sum_i \sum_j w_{ij} P_j \int_{F_i} dP(u|H_j) \qquad \text{(III.13)}$$

Assuming that the decisions from the peripheral sensors are independent conditioned on each hypothesis, (III.13) can be written as

$$\mathcal{R}_f' = \sum_i \sum_j w_{ij} P_j \int_{F_i} \prod_{k=1}^{N} dP(u_k|H_j) = \sum_i \sum_j w_{ij} P_j \prod_{k=1}^{N} \left(\int_{F_i} dP(u_k|H_j) \right) \qquad \text{(III.14)}$$

$$= w_{00} P_0 \prod_k \left(\int_{F_0} dP(u_k|H_0) \right) + w_{10} P_0 \prod_k \left(\int_{F_1} dP(u_k|H_1) \right)$$

$$+ w_{10} P_0 \prod_k \left(\int_{F_1} dP(u_k|H_0) \right) + w_{01} P_1 \prod_k \left(\int_{F_0} dP(u_k|H_1) \right)$$

$$+ w_{11} P_1 \prod_k \left(\int_{F_1} dP(u_k|H_1) \right) + w_{11} P_1 \prod_k \left(\int_{F_1} dP(u_k|H_1) \right)$$

$$= \int_{F_0} \left(w_{00} P_0\, dP(u|H_0) + w_{01} P_1\, dP(u|H_1) \right) + \int_{F_1} \left(w_{10} P_0\, dP(u|H_0) \right.$$

$$\left. + w_{11} P_1\, dP(u|H_1) \right) + \int_{F_2} \left(w_{20} P_0\, dP(u|H_0) + w_{21} P_1\, dP(u|H_1) \right)$$

(III.15)

The decision rule that minimizes the total risk assigns a particular combination of peripheral decisions u to that region that gives rise to the smallest integrand. Assuming that $w_{jj} = 0$, i.e. that there is no penalty for deciding correctly (a reasonable assumption in evidence processing), and that $w_{ij} - w_{2j} > 0$ for every j, i.e. that the cost of indecision is lower than the cost of deciding incorrectly, the test at the fusion becomes

$$\Lambda(u) \underset{d_0 \text{ or } d_1}{\overset{d_1 \text{ or } d_1}{\gtrless}} \frac{P_0}{P_1}\,\frac{w_{10}}{w_{01}} \qquad \text{(III.16)}$$

Similarly,

$$\Lambda(u) \underset{d_0 \text{ or } d_1}{\overset{d_1 \text{ or } d_1}{\gtrless}} \frac{P_0}{P_1}\,\frac{w_{10}}{w_{01} - w_{11}} \qquad \text{(III.17)}$$

and

$$\Lambda(u) \underset{d_0 \text{ or } d_1}{\overset{d_1 \text{ or } d_1}{\gtrless}} \frac{P_0}{P_1}\,\frac{w_{10} - w_{00}}{w_{01}}$$

(III.18)

where

$$\Lambda(u) = \frac{dP(u|H_1)}{dP(u|H_0)} = \frac{dP(u_1|H_1)\ldots dP(u_N|H_1)}{dP(u_1|H_0)\ldots dP(u_N|H_0)}$$

$$= \prod_{i\in S_1} \frac{P_{Di}}{P_{Fi}} \prod_{j\in S_0} \frac{1-P_{Dj}-P_{I'j}}{1-P_{Fj}-P_{I''j}} \prod_{n\in S_2} \frac{P_{I'n}}{P_{I''n}} \qquad \text{(III.19)}$$

where $P_{I'_j}$, $P_{I''_j}$ indicate the probability masses at sensor j associated with the fuzzy decision (or, indecision) d_{0v1} under the hypotheses H_1 and H_0 respectively, S_1 is the set of those decisions from the set u which favor $d_1 (= H_1)$, S_0 is the set that favors d_0 (= H_0), and S_2 is the set from the peripheral decisions u that favors d_2 (= {H_0 or H_1}, i.e. the undecided). Naturally, $u = S_0 + S_1 + S_2$.

From (III.16), (III.17), and (III.18), it follows that the optimal decision rule at the fusion is a likelihood ratio test. If we look at equation (III.19), the distribution of the LR under the two hypotheses is given by

$$P(\log\Lambda(u)|H_i) = P(\log\Lambda(u_1)|H_i) * \ldots * P(\log\Lambda(u_N)|H_i)$$

$$;\ i = 0, 1 \qquad \text{(III.20)}$$

where "*" indicates convolution, with

$$P(\log\Lambda(u_i)|H_0) = (1-P_{Fi}-P_{I''i})\,\delta\!\left(\log\Lambda(u_i) - \log\frac{1-P_{Di}-P_{I'i}}{1-P_{Fi}-P_{I''i}}\right)$$

$$+ P_{I''i}\,\delta\!\left(\log\Lambda(u_i) - \log\frac{P_{I'i}}{P_{I''i}}\right)$$

$$+ P_{Fi}\,\delta\!\left(\log\Lambda(u_i) - \log\frac{P_{Di}}{P_{Fi}}\right) \qquad \text{(III.21)}$$

and $\delta(x)$ is the Kronecker's delta function, i.e. $\delta(x) = 1$ if $x > 0$ and zero otherwise, and

$$P(\log\Lambda(u_i)|H_1) = (1-P_{Di}-P_{I'i})\,\delta\!\left(\log\Lambda(u_i) - \log\frac{1-P_{Di}-P_{I'i}}{1-P_{Fi}-P_{I''i}}\right)$$

$$+ P_{I'i}\,\delta\!\left(\log\Lambda(u_i) - \log\frac{P_{I'i}}{P_{I''i}}\right)$$

$$+ P_{Di}\,\delta\!\left(\log\Lambda(u_i) - \log\frac{P_{Di}}{P_{Fi}}\right) \qquad \text{(III.22)}$$

Hence, the distribution of $\Lambda(u)$ under H_0 is given as the product of all possible combinations of {(1 - P_{Fi} - $P_{I''i}$), $P_{I''i}$, P_{Fi}} according to their abscissae [9]. Similarly, the distribution of $\Lambda(u)$ under H_1 is given as the product of all possible combinations of {(1 - P_{Di} - $P_{I'i}$), $P_{I'i}$, P_{Di}} according to their abscissae. Then, the fusion is done by using the appropriate thresholds from (III.16), (III.17), and (III.18). In that case, the probability masses (or beliefs) associated with each decision are combined according to the threshold and their abscissae. Thus, the combining rule involves pairwise multiplication of probability masses according to Table II as in D-S theory. However, in GEP theory, the masses are associated via thresholds in an optimal way so that a certain risk (Eq. (III.15)) is minimized, or so that the probability of detection is maximized for fixed false alarm and indecision probabilities (generalized Neyman-Pearson test), whereas in D-S theory the probability masses (beliefs) are combined according to intersection of events, resulting in evidence conflict (see Section 3).

The probabilities in Table II are conditioned on each hypothesis, and $i = 0, 1$. Thus, each m_j^i, $j = 1, 2$, in Table II is a conditional probability for $i = 0, 1$. Hence, the initial probability combining takes place among conditional probabilities only. For $i = 0, 1$, each product term in Table I, is a probability mass on the LRT coordinate axis with abscissa $m_j^1(d) / m_j^0(d)$ for every $d = d_0, d_1, d_2$. Evidence combining under each hypothesis is done by summing the probabilities from Table II whose abscissae fall in specific intervals specified either by an optimization criterion, or a certain desired performance. Hence, for $d = d_0, d_1, d_2, \ldots, d_N$, evidence combining under each hypothesis H_i, $i = 0, 1$, is done according to the **threshold rule**

$$m_1^i(d_k)\, m_2^i(d_m) \rightarrow \text{decision } d_j \quad \text{if} \quad \frac{m_1^1(d_k)\, m_2^1(d_m)}{m_1^0(d_k)\, m_2^0(d_m)} \in F_j \qquad \text{(III.23)}$$

where F_j is the decision region that favors decision d_j. The regions F_j may be determined so that a performance criterion is optimized at the fusion (and possibly at the sensors). For a single binary hypothesis, the decision regions at the fusion are determined by simple thresholds, in which case the decision rule (III.23) simplifies to

$$m_1^i(d_k)\, m_2^i(d_m) \rightarrow \text{decision } d_j \quad \text{if} \quad t_j \le \frac{m_1^1(d_k)\, m_2^1(d_m)}{m_1^0(d_k)\, m_2^0(d_m)} < t_{j+1} \qquad \text{(III.24)}$$

for all k, m, and j, where t_j are the thresholds of the LRT's associated with the different decisions that minimize some risk function.

If **multiple hypotheses** (more than two) are tested, the combining rule is extended to combine the belief functions of the individual sources at the fusion and generate the new conditional belief function under each hypothesis. The association of the new belief function at the fusion with the set of admissible decisions must be done by using the multiple-hypotheses LRT [12], or another test that optimizes some performance measure. It must be underlined again, that the probabilities in the GEP combining rule need not be defined through Bayesian reasoning necessarily, but may very well correspond to belief functions resulting from the D-S approach.

Table II GEP Evidence Combining Rule

S2 / S1	$m_2^i(d_{\,})$	$m_2^i(d_{\,})$	$m_2^i(d_{\,})$
$m_1^i(d_{\,})$	$m_1^i(d_{\,})\, m_2^i(d_{\,})$	$m_1^i(d_{\,})\, m_2^i(d_{\,})$	$m_1^i(d_{\,})\, m_2^i(d_{\,})$
$m_1^i(d_{\,})$	$m_1^i(d_{\,})\, m_2^i(d_{\,})$	$m_1^i(d_{\,})\, m_2^i(d_{\,})$	$m_1^i(d_{\,})\, m_2^i(d_{\,})$
$m_1^i(d_{\,})$	$m_1^i(d_{\,})\, m_2^i(d_{\,})$	$m_1^i(d_{\,})\, m_2^i(d_{\,})$	$m_1^i(d_{\,})\, m_2^i(d_{\,})$

In the multiple hypotheses case, the conditional belief function in GEP becomes a multi-variable function of the LRs $(\Lambda_k(d) := \prod_{j=1}^{J} \frac{dP(d_j \mid H_k)}{dP(d_j \mid H_0)}, k = 1, 2, \ldots, m-1)$ where J is the number of sensors in the fusion system, d_j the decision of the j-th sensor, and m the number of tested hypotheses. The evidence from the different sensors is combined by forming the joint probability distribution of the LR's under each hypothesis, i.e. by generating $dP(\Lambda_1, \Lambda_2, \ldots, \Lambda_{m-1} \mid H_k)$, $k = 1, 2, \ldots, J$. For two sensors with independent decisions conditioned on each hypothesis, the conditional evidence combining rule of GEP for three hypothesis and soft decisions (fuzzy logic), can be implemented using Table III.

Once all the entries in Table III are entered, the evidence is *combined by adding the probabilities from the fourth column together when the corresponding abscissae, i.e. the pairs* $(\Lambda_1(d_1, d_2), \Lambda_2(d_1, d_2))$ *in the* second and third columns, are identical. Once the evidence from all sensors is combined using tables similar to Table III, decisions are associated with the combined evidence using rule (III.23) so that a desired performance criterion is optimized.

Thus, evidence combining at the fusion is done conditioned on each hypothesis separately. The evidence is then associated with the admissible decisions unconditionally using a LRT or a test that optimizes some performance measure. Notice that the set of decisions need not be the same as the set of hypotheses. Thus, **evidence combining and decision making are understood as separate concepts in the framework of the GEP theory**.

Table III Evidence combining rule for multiple hypotheses in GEP theory

(d_1, d_2)	$\Lambda_1(d_1, d_2)$	$\Lambda_2(d_1, d_2)$	$dP(\Lambda_1(d_1, d_2), \Lambda_2(d_1, d_2) \mid H_k)$ $= dP(\Lambda_1(d_1, d_2) \mid H_k) dP(\Lambda_2(d_1, d_2) \mid H_k)$ $= \prod_{j=1}^{2} dP(\Lambda_1(d_j) \mid H_k) dP(\Lambda_2(d_j) \mid H_k)$
(0, 0)	$\Lambda_1(0,0)$	$\Lambda_2(0,0)$	$dP(\Lambda_1(0,0) \mid H_k) dP(\Lambda_2(0,0) \mid H_k)$
(0, 1)	$\Lambda_1(0,1)$	$\Lambda_2(0,1)$	$dP(\Lambda_1(0,1) \mid H_k) dP(\Lambda_2(0,1) \mid H_k)$
(0, 2)	$\Lambda_1(0,2)$	$\Lambda_2(0,2)$	$dP(\Lambda_1(0,2) \mid H_k) dP(\Lambda_2(0,2) \mid H_k)$
(0, 0v1)	$\Lambda_1(0,0v1)$	$\Lambda_2(0,0v1)$	$dP(\Lambda_1(0,0v1) \mid H_k) dP(\Lambda_2(0,0v1) \mid H_k)$
(0, 0v2)	$\Lambda_1(0,0v2)$	$\Lambda_2(0,0v2)$	$dP(\Lambda_1(0,0v2) \mid H_k) dP(\Lambda_2(0,0v2) \mid H_k)$
(0v1, 0)	$\Lambda_1(0v1,0)$	$\Lambda_2(0v1,0)$	$dP(\Lambda_1(0v1,0) \mid H_k) dP(\Lambda_2(0v1,0) \mid H_k)$
(0v1, 1)	$\Lambda_1(0v1,1)$	$\Lambda_2(0v1,1)$	$dP(\Lambda_1(0v1,1) \mid H_k) dP(\Lambda_2(0v1,1) \mid H_k)$
(0v1, 2)	$\Lambda_1(0v1,2)$	$\Lambda_2(0v1,2)$	$dP(\Lambda_1(0v1,2) \mid H_k) dP(\Lambda_2(0v1,2) \mid H_k)$
(0v1, 0v1)	$\Lambda_1(0v1,0v1)$	$\Lambda_2(0v1,0v1)$	$dP(\Lambda_1(0v1,0v1) \mid H_k) dP(\Lambda_2(0v1,0v1) \mid H_k)$
(0v1, 0v2)	$\Lambda_1(0v1,0v2)$	$\Lambda_2(0v1,0v2)$	$dP(\Lambda_1(0v1,0v2) \mid H_k) dP(\Lambda_2(0v1,0v2) \mid H_k)$
(0v2, 1)	$\Lambda_1(0v2,1)$	$\Lambda_2(0v2,1)$	$dP(\Lambda_1(0v2,1) \mid H_k) dP(\Lambda_2(0v2,1) \mid H_k)$
... etc			

The generalization of the Bayesian (and N-P) theory by the GEP theory is straightforward. An interpretation is probably required to establish the correspondence between GEP and D-S theories. If the probabilities $P(u_k = i \mid H_j)$, $i = 1, 2, 3$, are considered as (conditional) bpa's (basic probability assignments [3]) in the D-S theory for the k-th sensor, $k = 1, 2, \ldots, N$, under hypothesis H_j, $j = 0, 1$, the evidence from the different sensors at the fusion is combined using the conditional distribution of the LR under the different hypothesis according to Table II or III. A new (conditional) belief function is generated using the decision thresholds at the fusion. The (hard) decisions at the sensors are used to simply produce a hard decision at the fusion, if needed, according to some optimality criteria. In that respect, the GEP theory not only defines and processes the evidence according to an a-priori set of optimality criteria, but also provides, if needed, for optimized hard decisions both at the local (sensor) as well as global (fusion) level, a capability which is not built-in the D-S theory (see Section 3).

The decision boundaries in GEP theory determine how evidence is associated with propositions at the fusion and reflect the choice of the costs w_{ij}. To demonstrate the effect that the semantic content of the local decisions has on the global decision (fusion), several experiments were conducted in Gaussian and *slow-fading* Rayleigh channels. The following statistical model were assumed for the two channels.

Gaussian: *Observation model at each sensor:* $r \sim G(0,1) : H_0$, and $r \sim G(s,1) : H_1$, where $G(\alpha,\beta)$ designates an α mean and variance β Gaussian distribution. If P_F is the operating false alarm probability, the associated threshold $t_D := Q^{-1}(P_f)$, where $Q(\ ) = 1 - \Phi(\ )$ is the cumulative distribution function (cdf) of the standard normal, and Q^{-1} is its inverse.

Rayleigh: *False alarm probability:* $P_F = [\lambda(1+\epsilon)]^{-(1+\frac{1}{\epsilon})}$;

Detection probability: $P_D = [P_F]^{(\frac{1}{1+\epsilon})}$

where λ is the threshold used, and ϵ the SNR at the sensor. In the single-level local logic Bayesian DDF with hard decisions at the sensors and fusion, the probabilities at the sensors were generated assuming

fixed false alarm probabilities at eh sensors equal to 0.05. For the multi-level local logic DDF, the ambiguous (soft or "fuzzy") decisions were generated by considering a ±20% uncertainty region about the thresholds that determine the decision boundaries in the Bayesian case. The numerical results that are presented refer to binary hypothesis testing with the set of "soft" decisions consisting of $\{d_i = H_0, d_i = H_1, d_i = H_0 vH_1\}$. Additional results for ternary hypothesis testing and arbitrary probability assignments can be found in [17], [16].

In Fig.s 8 and 9, the total single- and multiple-level logic (GEP) Bayesian DDF risk for the three cost assignments discussed earlier is plotted for Gaussian and slow-fading Rayleigh channels respectively. It is seen that for cases 1.1 and 1.2, the total risk is reduced when soft decisions (GEP) are used instead of hard decisions (Bayesian). Furthermore, the risk for the two cases decreases as the signal-to-noise ratio (SNR) decreases, in agreement with our intuition about a "good" fusion system. However, for case 1.3, the performance of GEP is not always superior to Bayesian, and: (a) soft decisions (GEP) do not lead to better performance than hard decisions (Bayesian); and (b) the risk increases, in general, as the SNR increases against our "intuition." The behavior for these three cases can be explained from Figs 8 and 9. For cases 1.1 and 1.2 the decision boundaries seem intuitively justifiable. However, for the case 1.3, the decision boundaries are "counter-intuitive." Thus, given a set of cost factors, evidence combining in GEP theory is done according to a set of performance criteria, so that the performance of the fusion is optimized. The ability of GEP theory to process evidence independent of the associated propositions and combine it according to desired performance criteria without evidential conflict is highly desirable in sensor integration and decision fusion problems where we want to design a fusion system so that its performance can be assessed a priori and according to certain design criteria and specifications.

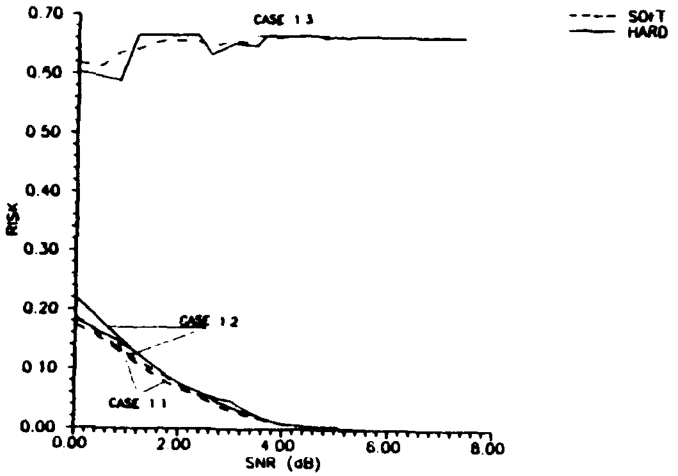


Fig. 8 Risk vs SNR. Gaussian case; 5 sensors.

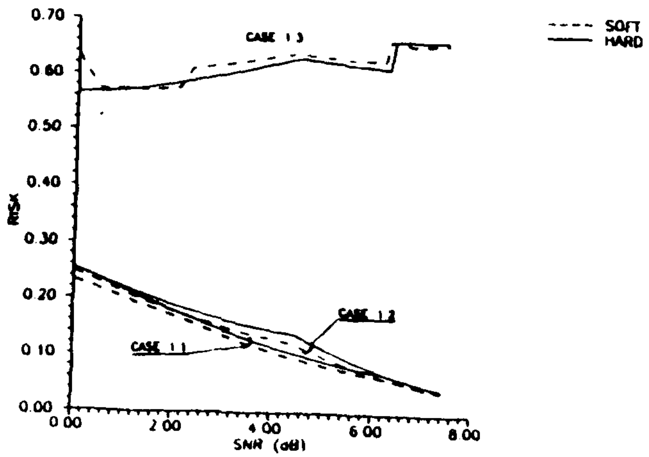
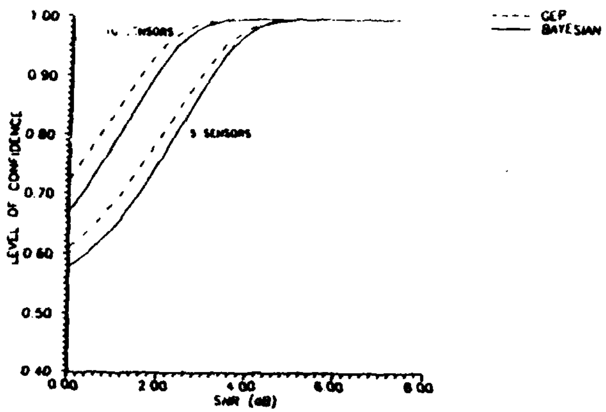
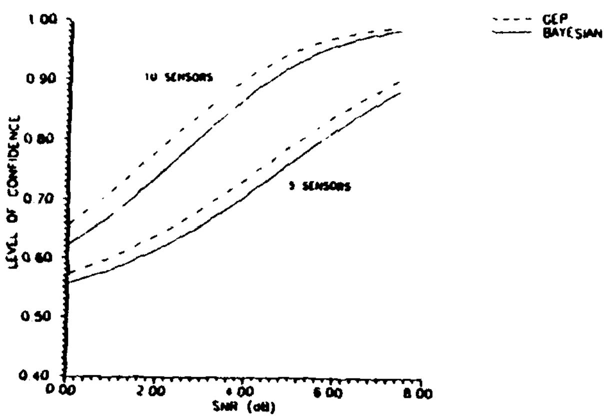


Fig. 9 Risk vs SNR. Rayleigh case; 5 sensors.

In another set of experiments on binary hypothesis testing with Gaussian and Rayleigh distributed data, the performance of the GEP fusion with soft decisions at the sensors and hard decisions at the fusion was compared to the Bayesian DDF with hard decisions at all sensors and the fusion. The fusion system that was used for comparison consisted of five and ten identical sensors. The sensors decisions were assumed statistically independent under either hypothesis. The soft decision "H_1 or H_0" was generated by assigning a "±20% uncertainty region" about the threshold that was used for the Bayesian DDF to generate hard decisions at the sensors at 0.05 false alarm probability level. The *Level Of Confidence* (*LOC*), which is equivalent to the (unconditional) probability of correct decision, was used for comparison. The LOC curves in Fig. 10 indicate that GEP outperforms Bayesian DDF with hard local decisions in all cases. The curves were obtained by assuming a fixed false alarm probability 0.05 at the sensors and 0.005 at the fusion.05. GEP outperforms hard-decision Bayesian *DFF in both binary and ternary* hypothesis testing, in both Gaussian and slow-fading Rayleigh channels and for any number of sensors. This does not come as a surprise if the decision set of GEP is thought of as the result of multi-level quantization of the data, and the quantization is done according to a semantically intuitive fashion. Similar conclusions were drawn from the comparison of GEP DDF with the Bayesian DDF in ternary hypothesis testing [17], [16].



(a) Gaussian case; $P_{F,sensor} = 0.05$; $P_{F,fusion} = 0.005$



(b) Rayleigh case; $P_{F,sensor} = 0.05$; $P_{F,fusion} = 0.005$

Fig. 10 Level of Confidence (LOC) vs. SNR for Bayesian and GEP decision fusion in binary hypothesis testing with (a) Gaussian, and (b) Rayleigh distributed data in five and ten sensor DDF system. *In all cases, the fusion makes hard decisions regarding the true nature of the tested hypothesis.* The LOC corresponds to the total probability of deciding correctly. *The solid curves correspond to Bayesian DDF. The dashed curves correspond to GEP fusion. In all cases, the GEP fusion outperforms the Bayesian fusion.*

IV. Distributed Decision Fusion using Dempster-Shafer's Theory

The difference between the Bayesian and D-S theory lies on the type of information that each sensor transmits to the fusion after *processing the* data locally. As it will become clear in the sequel, if the propositions in the D-S theory are identified with decisions in the GEP (Generalized Bayesian) theory, then there are no semantic differences *in the frame of* discernment between the two theories. The difference lies on that the frame of discernment in the GEP theory consists of decisions with assigned probabilities that satisfy the Bayesian rule, *whereas the frame* of discernment in the D-S theory consists of propositions that do not, in general, satisfy the Bayesian rule. *Assuming that the number of* hypotheses that are tested is fixed and the number of decisions (or frame of discernment in the D-S terminology) is fixed, the output of the local data processing is a set of probabilities regarding the likelihood that the data have been generated by one of the particular hypotheses or subset of hypotheses according to the frame of discernment. To *that extend, the use of the* term decisions in the D-S theory does *not precisely* reflect the output of the local processing. It is more appropriate to characterize the outcome of the local processing as evidence about a chosen *set of* proposition rather than decision regarding a specific hypothesis or set *of* hypotheses. Thus, even if the frame of discernment is kept common between Bayesian and D-S approaches (by utilizing multi-level Bayesian logic), the mapping of the data in the output of the local processor is completely different; the Bayesian processor maps each data to a particular, *single decision (integer-valued* scalar), whereas the D-S processor *maps* the same data to a set of probabilities (multidimensional real-valued *vector*) associated with all decisions in the frame of discernment. *Hence, the communication* requirements between Bayesian processors and fusion in one, and D-S processors and fusion on the other are different. Assuming a frame of discernment consisting of k propositions, the communication requirements for the Bayesian case is 2logk (the bandwidth required to transmit one of k bits), whereas for the D-S processor k analog outputs must be transmitted to the fusion. Thus, unless the communication requirements for the two approaches are made common, no direct comparison in the performance of the two schemes is meaningful. Since such a performance is beyond the objectives of this paper, we limit the discussion in the structure of the D-S DDF.

In D-S theory, a set of mutually exclusive and exhaustive propositions $u_1, u_2, \ldots, u_m$ is assumed toward which evidence is being offered. To each proposition, their disjunctions, and negations, a nonnegative number between zero and one (or probability mass) is assigned. If A is an atomic proposition, a disjunction of propositions, or a negation of a proposition, then a probability mass, m(A), is assigned to A. *The quantity m(A) is a measure of the belief in* proposition A based on the evidence offered. If U designates the frame of discernment, then

$$\sum_{A \in U} m(A) \le 1 \qquad \text{(IV.1)}$$

with the remaining $1 - \sum_{A \in U} m(A)$ mass attribute to ignorance. Assuming that ignorance constitutes a separate proposition and extending the set U to include this proposition, expression (IV.1) holds as an equality. According to D-S theory, a support function is defined for single propositions as

$$spt(u_i) = m(u_i) \qquad \text{(IV.2)}$$

and for more complex propositions as

$$spt(A) = \sum_{B \subset A} m(B) \qquad \text{(IV.3)}$$

where "C" indicates subset. The plausibility function is defined as

$$pls(u_i) = 1 - spt(\bar{u}_i) \qquad \text{(IV.4)}$$

where $\bar{u}_i$ indicates the negation of proposition u_i. Alternatively, the plausibility function for a proposition u_i is obtained by summing the masses of all the disjunctions that contain u_i, including itself, i.e.

$$pls(u_i) = \sum_{u_i \in A} m(A) \qquad \text{(IV.5)}$$

Hence, the support function is indicative of how much evidence is offered in support of a given proposition by all the propositions that relate to it. Furthermore, the plausibility function is indicative of how likely it is for a given proposition to have generated the data.

Evidence from different, and *independent*, sources defined over the same frame of discernment, is fused according to Dempster's combining rule [Depm '68]

$$m(u_i) = m_1 \oplus m_2 = \frac{\sum_{A_i B_j = u_i} m_1(A_i)\, m_2(B_j)}{1 - \sum_{A_k B_m = \phi} m_1(A_k)\, m_2(B_k)} \qquad \text{(IV.6)}$$

where m_1 and m_2 designate the support (belief) functions from the two different sources of evidence defined over the same frame of discernment, u_i is the proposition toward which evidence is sought, and "ϕ" is the empty set [Shaf '76]. Renormalization of the combined evidence in the rule (IV.6) is required to reject evidence that corresponds to conflicting propositions. The D-S combining rule can be implemented in a tabular fashion. *To illustrate the mechanical similarities that exist* between the Dempster's combining rule and the GEP DDF, consider a simple binary hypothesis testing problem. If the frame of discernment is defined as $(u = H, u = H, u = H$ or $H)$, with u, indicating the inability to associate evidence from the data with a definite hypothesis, the Dempster's combining rule for two sensors can be implemented using Table IV. In Table IV, k designates evidence associated with conflicting propositions which is used as normalizing factor in (IV.6). The combined evidence is calculated by summing all the product terms from Table IV that result to the same intersection proposition, and normalizing the result. In multiple-source evidence combining, rule (IV.6) is repeated sequencially until the evidence from all sources is exhausted.

The difference between the D-S and Bayesian theory is that the probability assignments for the propositions in the frame of discernment of the D-S theory do not satisfy the fundamental axiom of (Bayesian) probability, namely

$$P(A+B) = P(A) + P(B) - P(AB) \qquad (IV.7)$$

In the D-S context, the proposition A+B is viewed as a separate entity in the frame of discernment and can be assigned an arbitrary probability mass. Still all the probability assignments in the D-S theory must add up to one or some positive quantity less than one, with the remaining probability mass to add up to one attributed to total ignorance [Shaf '76]. A correspondence between the propositions as defined in the D-S theory and the decisions as defined in the multi-level logic Bayesian theory can be established if the decisions of the multi-level logic Bayesian framework are identified with the propositions in the D-S frame of discernment. *Once this correspondence is established the fusion performance under the two approaches can be studied under common*

communication constraints. By disassociating decisions from the hypotheses under test, the Generalized Evidence Processing (GEP) provides a semantically common framework within which the N-P and D-S DDF approaches can be compared under common communication constraints.

Table IV Dempster's Combining Rule.

S2 / S1	$m_2^1(u)$	$m_2^1(u)$	$m_2^1(u)$
$m_1(u)$	$m(u)=m_1(u)\,m_2(u)$	$k=m_1(u)\,m_2(u)$	$m(u)=m_1(u)\,m_2(u)$
$m_1(u)$	$k=m_1(u)\,m_2(u)$	$m(u)=m_1(u)\,m_2(u)$	$m(u)=m_1(u)\,m_2(u)$
$m_1(u)$	$m(u)=m_1(u)\,m_2(u)$	$m(u)=m_1(u)\,m_2(u)$	$m(u)=m_1(u)\,m_2(u)$

Due to the difference in the way evidence is generated in Bayesian (N-P) and D-S theory, an unconditional performance comparison between the two theories is not, in general, feasible. Since in a lot of practical applications the performance of a decision making system is determined by fixing the false alarm probability and maximizing the detection probability at the fusion, it is meaningful to compare the Bayesian and D-S approach based on an *N-P* criterion. *In order to make the comparison possible*, we assume that the basic probability assignment of the D-S DDF at the local level is determined using the likelihood function, i.e. we assume that

$$m(\alpha \mid r) = P(\alpha \mid r) \qquad (IV.8)$$

where a designates a proposition towards which evidence is provided, and r the observations. Even when the bpa is resolved at the local level, the decision rule at the fusion after the local evidence is combined remains undetermined. In order to keep the decision rule in a D-S context while maintaining a basis for comparison with the Bayesian DDF, the decision rule that will be used for the D-S DDF will assign the data to the proposition that has the highest support among all propositions in the frame of discernment that correspond to definite hypotheses, i.e.

$d(r) := d_i(r) : \max_i \, spt(d_i)$ and $d_i = H_i$, i over all single hypothesis propositions (IV.9)

With the above assumptions, we prove the following theorem.

Theorem 6 Assume that the objective of the fusion is to maximize the detection probability after fusion for fixed false alarm probability. Let the observations of the local sensors be independent from each other conditioned on each hypothesis. Let the bpa for the D-S DDF be determined by the likelihood function (IV.8) at the local level. If the fusion rule is the rule (IV.9) above, then:

(a) if the local frame of discernment coincides with the hypotheses under test, i.e. no unions of hypotheses are used as basic propositions, the performance of the D-S DDF is the same as the centralized N-P (Bayesian) fusion.

(b) if compound-hypotheses propositions are allowed in the local bpa, then the performance of the D-S DDF is always inferior to the centralized N-P fusion and the distributed N-P fusion for the same communication overhead.

Proof We prove the theorem for the case of two sensors and binary hypotheses testing. A generalization of the proof, although notationally involved, does not present any conceptual difficulties and as such is omitted.

Part (a) According to the assumptions of the theorem, the bpa is

$$m(H_i) := Pr(H_i \mid r) = [p(r \mid H_i)Pr(H_i)] / p(r) \; ; \; i = 0, 1 \qquad \text{(IV.10)}$$

and so the D-S requirement

$$m(H_0) + m(H_1) = 1 \qquad \text{(IV.11)}$$

is satisfied. Using the Dempster's combining rule (IV.6) for two sensors, we obtain

$$sup(H_1) = [m^1(H_1)m^2(H_1)] / [1 - m^1(H_1)m^2(H_0) - m^1(H_0)m^2(H_1)] \qquad \text{(IV.12)}$$

where the division is the result of renormalization due to the existence of conflicting evidence mass after fusion, and the superscripts identify the sensors. A similar expression is obtained for the H_0 hypothesis if the indexes in (IV.12) are switched. The proposed decision rule (IV.9) translates to

$$sup(H_1) / sup(H_0) \mathop{\gtrless}_{H_0}^{H_1} t \qquad \text{(IV.13)}$$

were t is some threshold to be determined. Taking into account that for this particular case the D-S rule yields

$$sup(H_i) = m(H_i) \qquad \text{(IV.14)}$$

and using expression (IV.3), the D-S decision rule gives after some elementary algebra

$$[p(r_1 \mid H_1)p(r_2 \mid H_1)] / [p(r_1 \mid H_0)p(r_2 \mid H_0)] \mathop{\gtrless}_{H_0}^{H_1} t \qquad \text{(IV.15a)}$$

or

$$[p(r_1 \mid H_1)p(r_2 \mid H_1)] / [p(r_1 \mid H_0)p(r_2 \mid H_0)] \mathop{\gtrless}_{H_0}^{H_1} t \qquad \text{(IV.15b)}$$

or

$$[p(r_i\,|H_1)/p(r_i\,|H_0)]\,[p(r_j\,|H_1)\,/\,p(r_j\,|H_0)]\ \underset{H_0}{\overset{H_1}{\gtrless}}\ t \qquad \text{(IV.15c)}$$

or

$$[\,p(r_i\,|H_1)p(r_j\,|H_1) - t\,p(r_i\,|H_0)p(r_j\,|H_0)]\ \underset{H_0}{\overset{H_1}{\gtrless}}\ 0 \qquad \text{(IV.15d)}$$

which is precisely the centralized Bayesian N-P test. Thus, the performance of the D-S DDF in this case is identical to the optimal centralized Bayesian DDF for the same false alarm probability at the fusion.

Part (b) In the binary hypotheses testing case the only compound proposition in the frame of discernment is $\{H_0$ or $H_1\}$. If we assume, without loss of generality, that the bpa for the three propositions is done by subtracting an equal amount of probability from the two propositions that correspond to the definite hypotheses and associating it with the compound proposition, the following bpa results

$$\begin{aligned} m_i(H_0) &= Pr(H_0\,|r_i) - \epsilon(r_i)/2 \\ m_i(H_1) &= Pr(H_1\,|r_i) - \epsilon(r_i)/2 \qquad \text{(IV.16)} \\ m_i(H_0 \text{ or } H_1) &= \epsilon(r_i) := \epsilon_i \end{aligned}$$

where the probability mass $\epsilon(r_i)$ can be data dependent. Using the Dempster's combining rule to fuse the evidence and suppressing the explicit dependence of ϵ_i on the data for notational simplicity, we obtain the following expressions for the support function regarding the two hypotheses.

$$\sup(H_0) = \frac{[\,m_1(H_0)m_2(H_0) + 1/2[\,\epsilon_1 m_2(H_0) + \epsilon_2 m_1(H_0)] - 3\epsilon_1\epsilon_2/4\,]}{[1 - \text{conflicting evidence}]} \qquad \text{(IV.17a)}$$

and

$$\sup(H_1) = \frac{[\,m_1(H_1)m_2(H_1) + 1/2[\,\epsilon_1 m_2(H_1) + \epsilon_2 m_1(H_1)] - 3\epsilon_1\epsilon_2/4\,]}{[1 - \text{conflicting evidence}]} \qquad \text{(IV.17b)}$$

from which the assumed decision rule

$$\sup(H_1)\,/\,\sup(H_0)\ \underset{H_0}{\overset{H_1}{\gtrless}}\ t \qquad \text{(IV.18)}$$

yields

$$\begin{aligned} &(\,p(r_1\,|H_1)p(r_2\,|H_1) - t\,p(r_1\,|H_0)p(r_2\,|H_0)) \\ &\quad +1/2(\,(\epsilon_2\,p(r_1\,|H_1) + \epsilon_1\,p(r_2\,|H_1)) \\ &\quad - t(\,p(r_1\,|H_0) + p(r_2\,|H_0))\,)\ \underset{H_0}{\overset{H_1}{\gtrless}}\ 3\epsilon_1\epsilon_2/4(1-t) \qquad \text{(IV.19)} \end{aligned}$$

By comparing the decision rule (IV.19) with the optimal N-P test rule (IV.15d), it is seen that the first term in brackets in the left side of (IV.19) is identical to the term in the left side of (IV.15d). Since the decision rule (IV.15d) is *the* optimal decision rule in the N-P sense, rule (IV.19) would achieve optimal performance *if and only if* the rest of the terms in (IV.19) could be made identically equal to zero for a fixed threshold t. However, even with data dependent bpa assignment $\epsilon_i(r)$, this is not possible in general. Thus, the performance of the D-S DDF is inferior to the optimal centralized N-P fusion. Furthermore, since the performance of the distributed N-P decision fusion can be arbitrarily close to the optimal centralized one [TVB '87, Thom '90] by simply including some additional quality information bits along with the decisions or by increasing the number of quantization levels, the performance of the N-P DDF is always superior to the performance of the D-S DDF for a lesser amount of communication requirements. Notice that in the D-S either the data itself has to be transmitted from the sensors to the fusion (which is the most efficient way), or the bpas must be transmitted thus making the communication requirements proportional to the number of propositions in the frame of discernment. (Clearly, a quantized version of the data or bpas can be transmitted resulting in reduction of communication requirements and performance as well.)

The above arguments extend easily to multiple sensor case. The general multi-hypothesis case can be handled in a similar way as the two hypothesis case, only the expressions become more complicated. □

In order to compare the consistency of the GEP and D-S evidence combining rules (III.23,24) and (IV.6) respectively, the following experiment was conducted. Two identical sets of propositions were used for the GEP and D-S DDF. The basic probability assignment for the supported propositions in the D-S fusion was taken to be identical to the associated likelihood fusnction (conditional probability) that was used in the GEP fusion according to (IV.8). The likelihood function was calculated using the "uncertainty regions" about the thresholds that would correspond to a hard decision Bayesian DDF. The details of the construction of the likelihood function are discussed in length in [17]. The experiment was conducted for binary and ternary hypothesis testing fusion systems, and numerical results have been obtained for distribution based, as well as arbitrary, bpas. However, only results from the binary hypothesis testing will be presented. For additional results, the reader is referred to [ThGa '90 and Ga '90]. For the GEP fusion and the binary hypothesis testing, the conditional probabilities at the sensors were obtained by associating the "±20% uncertainty region" around the threshold that corresponded to seansor false alarm probability 0.05 with the ambiguous decision {H_0 or H_1}. The so obtained conditional probabilities were used as the original probability assignments at the sensor for the D-S fusion.

The conditional probability masses were calculated at the fusion using Dempster's combining rule. The conditional probabilities that resulted from the GEP fusion and the conditional probability masses from D-S fusion were then used to calculate conditional plausibility according to (IV.5). The results were obtained for a false alarm probability of .05 at the sensor and .005 at fusion.

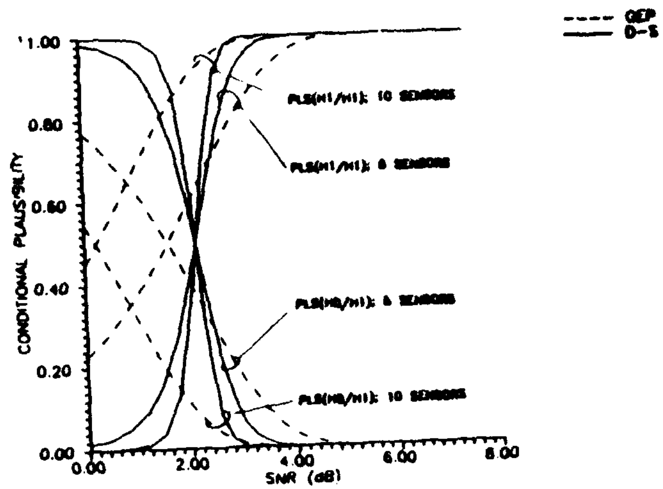


Fig. 11 Conditional plausibility for five and ten sensors. Gaussian case. Sensor false alarm $P_{F_s} = 0.05$; False alarm after fusion $P_{F_f} = 0.005$.

Figures 11 and 12 display results for Gaussian and Rayleigh distributed signals respectively. Both graphs show the plausibility conditioned on hypothesis H_1 for five and ten sensors. To compare the two combining rules for consistency, we define the crossover point as the

SNR level above which the plausibility for the correct hypothesis, H_1, becomes greater than that for the incorrect hypothesis, H_0. Observe that for both the five and ten sensor cases the crossover point occurs at a lower SNR for GEP than for D-S theory. So GEP works correctly for a wider range of SNR than does D-S theory. Also notice the behavior as the number of sensors increases from five to ten. For GEP the crossover point moves to lower SNR while for D-S theory it does not move at all. This indicates that we can improve the performance of GEP by increasing the number of sensors, which is a very desirable feature. The performance of D-S theory, on the other hand does not improve when the number of sensors increases.

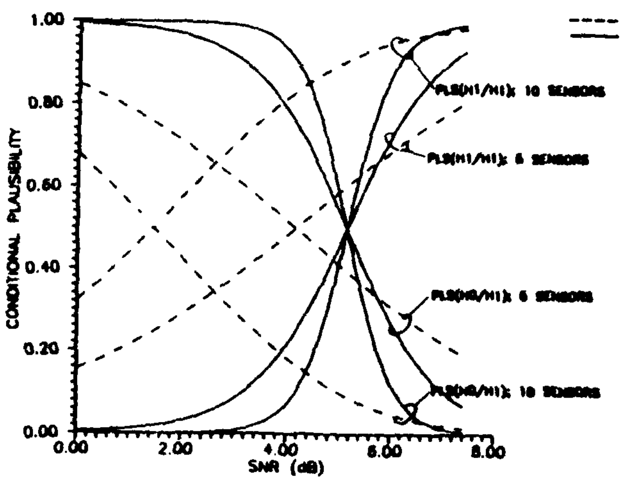


Fig. 12 Conditional plausibility for five and ten sensors. Rayleigh case. Sensor false alarm $P_{F_s} = 0.05$; False alarm after fusion $P_{F_f} = 0.005$

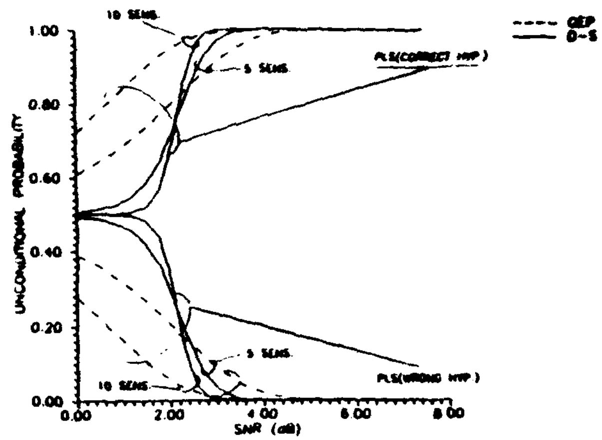


Fig. 13 Unconditional plausibility vs. SNR. Gaussian case. Sensor false alarm $P_{F_s} = 0.05$; False alarm after fusion $P_{F_f} = 0.005$

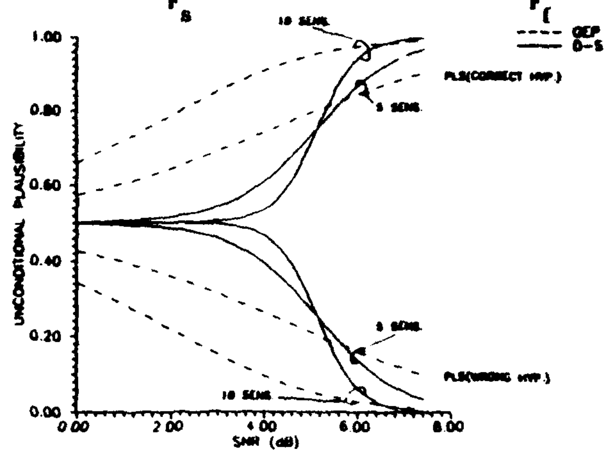


Fig. 14 Unconditional plausibility vs. SNR. Rayleigh case. Sensor false alarm $P_{F_s} = 0.05$; False alarm after fusion $P_{F_f} = 0.005$

Figures 13, 14 show unconditional plausibility plots for the Gaussian and Rayleigh cases. More specifically they show the unconditional plausibility for the correct and incorrect hypotheses. Once again the results are shown for both five and ten sensors. We see that for all cases the plausibility for the correct hypothesis is higher at lower SNR for GEP than that for D-S theory. The separation between plausibility for correct and incorrect hypotheses is much clearer for GEP. In fact at very low SNR D-S theory fails to separate the plausibility for the correct hypothesis from that of the incorrect.

V. Bayesian / N-P DDF and Neural Networks

Although, the form of the optimal Bayesian / N-P DDF is known, for both binary and multi-level quantizations (Theorems 1 and 5), the optimal thresholds are given, in general, in terms of coupled, nonlinear equations [8], [10], whose solution is not forthcoming, even in simple cases. Suboptimal numerical solutions to the N-P DDF [10] may still be computationally intensive, if the fusion rule is unknown. The optimal solution to the Bayesian and Neyman-Pearson DDF problem, Eqs. (II.3c) and (II.4d) respectively, bares striking topological and functional similarities with the structure of a neural network (NN). This topological similarity suggests an alternative approach to solving the computationally N-P hard [5] DDF problem. By slightly modifying the values that designate the decision at the i-th sensor to

$$u_i = \begin{cases} +1 \text{ if the i-th local decison favors hypothesis } H_1 \\ 0 \text{ if the i-th local decison favors hypothesis } H_0 \end{cases} \qquad \text{(V.1)}$$

for notational convenience, the optimal Bayesian and N-P DDF rules (II.3a) and (II.3b) respectively take on the form

$$\sum_i (w_i u_i + t_i) \underset{H_0}{\overset{H_1}{\gtrless}} T_f \qquad \text{(V.2)}$$

where

$$w_i = \log\left(\frac{P_{Di}}{P_{Fi}}\right) - \log\left(\frac{1-P_{Di}}{1-P_{Fi}}\right) \qquad \text{(V.3)}$$

and

$$t_i = \log\left(\frac{1-P_{Di}}{1-P_{Fi}}\right) \qquad \text{(V.4)}$$

By combining the constant thresholds together with the unknown operational threshold T_f, and defining

$$w_0 := -T_f + \sum_i t_i \qquad \text{(V.5)}$$

the DDF rule (V.2) can be written in a form reminiscent of an NN architecture:

$$w_0 + \sum_i w_i u_i \underset{H_0}{\overset{H_1}{\gtrless}} 0 \qquad \text{(V.6)}$$

A noticeable advantage of (V.6) over (V.2) is that the unknown threshold T_f has been absorbed in the synaptic weight w_0, which can be determined through training by assuming that it correspond to the interconnection weight of an additional, constant input to the fusion neuron. Notice that the threshold in (V.6) is known, constant, and equal to zero. Thus, (V.6) can be implemented using an NN and replacing the hard threshold decision rule by a smoother sigmoidal nonlinearity [McRu '87, Nils '90, TPS '91].

In Fig. 15 the optimal Bayesian (N-P) DDF structure is shown when the local LR is linear on the data. If the (local) sensors and fusion in

Fig. 15 are identified with neurons and the thresholds are replaced by continuous sigmoid functions, there is a one-to-one topological correspondence between the D-S DDF architecture and the simple, two layer NN of Fig. 16. The topological similarities suggest that one can take advantage of the learning capabilities of an NN and train it to solve the Bayesian DDF even when the channel statistics are not known. The solution to Bayesian DDF can be achieved by using any of the available training rules. For example, if a quadratic error is defined at the fusion by squaring the difference between the actual hypothesis and the output of the fusion and using a gradient based algorithm, such as backpropagation [20], to update the synapses weights, i. e. the coefficients of the LRTs in the Bayesian DDF.

Training of the NN with a quadratic error criterion will result in a minimum error computer, if trained properly. A quadratic error training criterion attempts to fit the data in two different hypothesis

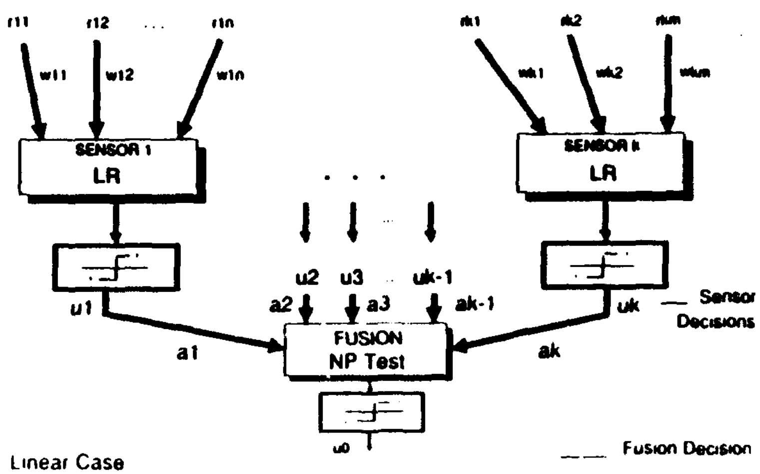


Fig. 15 Optimal Linear Bayesian DDF Configuration

by minimizing a distance criterion. However, if the data in the training set are numerically close under the two hypothesis, overtraining of the NN in order to achieve perfect discrimination of the data in the training set, will result in poor post-training performance. To avoid performance degradation from overtraining, selective training has been used in [22] with excellent results. The drawbacks associated with overtraining in the quadratic error criterion can be avoided by using an N-P based optimality criterion, such as the minimization of the miss probability at the fusion for fixed false alarm probability [22]. Such a training criterion results in an NN that implements the optimal N-P DDF. If the optimal Bayesian DDF is highly nonlinear, an NN can be used to solve the optimal Bayesian DDF by considering a truncated Taylor's series expansion of the LRT (or a Voltera series), and an NN similar to the one in Fig. 16 for determining the coefficient for each power in the T.S.E. [22].

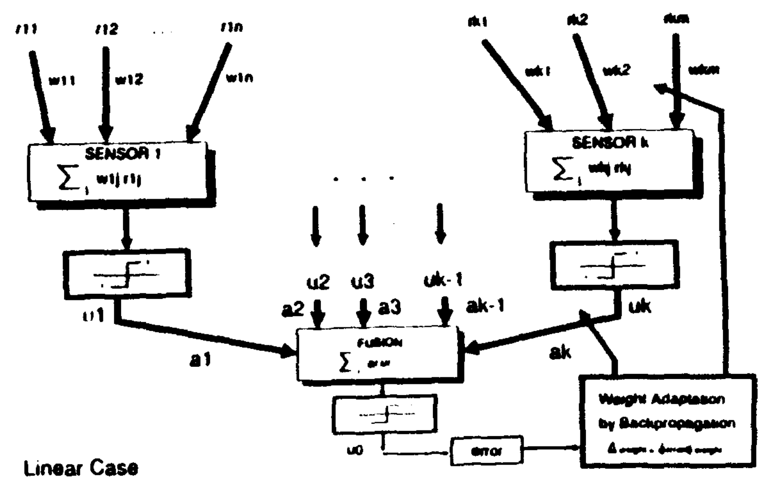


Fig. 16 Equivalent Neural Network for Bayesian DDF

A. TRAINING RULES

1. Backpropagation based on mean-squared error

Let the training output of the network be u_0^n at the n-th iteration, while the training hypothesis is u_i^n. The standard backpropagation method trains the NN by minimizing the

$$\text{error energy } \Sigma_n (u_0^n - u_i^n)^2. \quad (V.7)$$

To apply the generalized delta rule [McRu '86], define for each neuron k the function

$$\delta_k = o_k(1-o_k)\Sigma_{\text{all } j \text{ that } k \text{ leads to}} \delta_k w_{kj}\,. \quad (V.8)$$

where o_j is the output of neuron j and w_{kj} is the current weight between node k and node j. The output node is a special case where

$$\delta_n = 2*(u_0^n - u_i^n)*u_0^n*(1-u_0^n). \quad (V.9)$$

The update of the weights during training is done using the difference equation

$$dw_{ij}^n = \eta\delta_j o_i + \alpha\, dw_{ij}^{n-1}. \quad (V.10)$$

where η and α are predefined constants that determine the rate of convergence. The second term in the weight update equation is known as the momentum term.

Backpropagation was used to train a neural network to perform DDF. The network consisted of four identical sensors and a fusion. Each sensor and the fusion were represented by identical NN's, each having two input neurons, one hidden layer with three neurons, and a single-neuron output layer. The NN was first trained on the binary hypothesis problem H_0 : w and H_1 = A + w, where w is a zero mean, unit variance Gaussian random variable, and A is a constant. The backpropagation algorithm

converged for cases where A = 0.5, 1, and 3, and produced the post-training Receiver Operating Characteristics (ROCs) shown in Fig.s 17 a, b, and c. A comparison with the optimal fusion ROCs, indicates a very close matching between the ROCs obtained by the NN and the optimal ones [10].

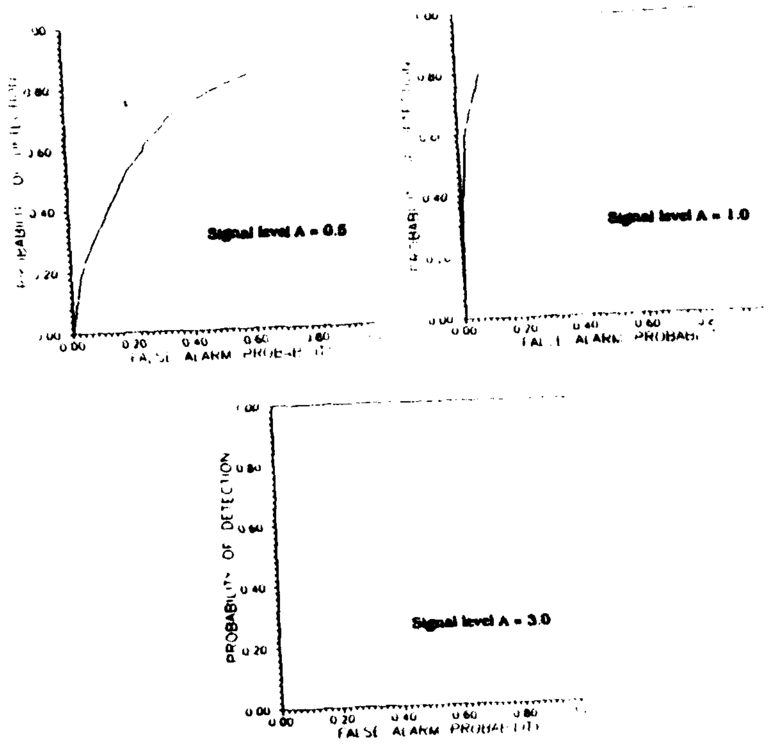


Fig. 17 Receiver Operating Characteristics obtained by the NN for different signal levels; Gaussian case. In all cases, the noise variance is set equal to one.

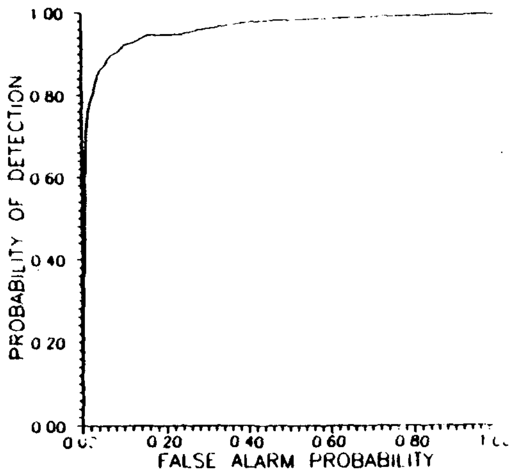


Fig. 18 NN DDF Receiver Operating Characteristics for Rayleigh case;
Variance under H_0 = 1; Variance under H_1 = 4.

To test the effectiveness of the NN DDF solution in non-gaussian data, the NN was trained using Rayleigh distributed data with variance one under H_0, and variance four under H_1. The ROC that was obtained after training and shown in *Fig. 18 is very close to the optimal one* [10]. The ROCs in all cases were obtained by using 9800 sample data other than the training data set and externally varying the threshold w_0 in (V.7) to obtained the entire range of (P_F, P_D).

The training was done as follows: A set of 100 training samples were generated, 50 from each hypothesis, and repeatedly presented in the NN until convergence. The test for convergence was based on the criterion

$$\sqrt{\left(\Sigma_{n=1}^{100}\Sigma_{ij \text{ all weights}}(dw_{ij}^n)^2 + \Sigma_{ij \text{ all weights}}(w_{ij}^n)^2\right)} \qquad \text{(V.11)}$$

Training was terminated when the criterion (V.11) was satisfied.

Initially, all samples were used during training. After a number of iterations though, samples producing the wrong result (i.e. giving a difference more that 0.5 in absolute value) were ignored (not discarded!) and trained was not used on them, i.e. selective training was used. Training was finished within five or six iterations after selective training was introduced. However, when selective training was introduced, was found to be critical.

At first, the network did not differentiate between H_0 and H_1. It responded to the training by driving the output for both cases down after H_0, and up for both cases after H_1. Both cases produce approximately the same output at each case for a number of iterations. After that number, differentiation starts and each result converges separately.

Starting *the selective training* before the network starts to learn selecting between the two hypotheses would mean that one of the hypotheses is screened out (the one further from the common current output). This would then imply that from this point on, the network sees a uniform output 0 as the desired result, and effectively then approximates the *zero function*, the simplest function giving this result, making it effectively useless for recognition.

If selective training is started after differentiation is made, then both cases are represented and the network converges very fast. The point at which selective training can start was found to be after 3 iterations for A = 3, after about 50 iterations for A =1, and after more than 90 iterations for A = 0.5, consistent with the difficulty in discriminating among the data from the two hypotheses in the three cases

(Note: one iteration designates a complete training cycle with all data from the training set.)

Although selective training expedited convergence of the NN, it did not seem to affect the final performance of the NN substantially. However, as it was mentioned above, it was found to be of critical importance when selective training started. This sensitivity is due to the fact that the NN was initially assumed untrained. Thus, the decisions made by the individual sensors might be completely erroneous. One way to avoid this sensitive behavior is to either pre-train the NN that corresponds to each sensor separately, and then retrain them all together in the fused NN, or used persistent training at each training sample presentation.

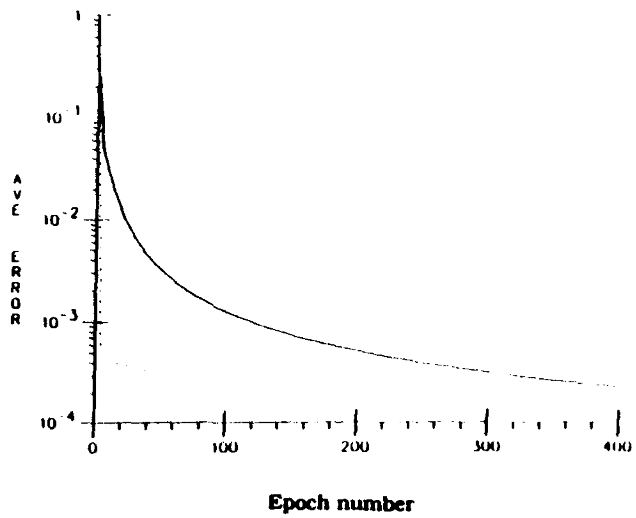


Fig. 19 Average fusion error per data set vs. epochs during training

Solid line: Permissible error/data/epoch = 0.1

Dotted line: Permissible error/data/epoch = 0.001

Simulations of the NN solution with persistent training per training sample have shown fast convergence with relatively small size training data in the linear Bayesian DDF problem. An NN for a DDF case of five sensors, each receiving five i.i.d. samples from a Gaussian distribution with zero mean under hypothesis H_0, mean one under hypothesis H_1, and identical standard deviations equal to two under both hypotheses, was simulated using a set of one hundred training points. For each training point, backpropagation was used to update the synapses. At each learning cycle, scores of correct classification and false alarms at first presentation were kept, so that the probability of false alarm was adjusted to the desired level. Otherwise, weight update took place through backpropagation until the deviation between the true hypothesis and the NN output was within prescribed tolerance limits.

The simulation results that are presented below correspond to binary hypothesis testing, equal a priori probabilities for each hypothesis, and five identical sensors. Five observations were made at each sensor. To train the NN, 30 training data sets were generated and fed in. Using back-propagation, the weights of the synapses were adjusted when the error at the fusion exceeded the prescribed level of tolerance until the fusion error was reduced to prescribed level of tolerance. Fig. 19 shows the converging behavior of the NN during training for different allowed error tolerance. Cases with different learning rates were also simulated and the results are shown in Fig. 20. It is seen that the average error at the fusion as function of the number of epochs (training cycles) reduces faster in the case of larger learning rate. At the end of training, the synaptic weights were fixed at their values at the end of the last training cycle.

The post-training performance of the NN was evaluated using 2000 testing data sets that were generated under the equal-likely hypothesis assumption. By varying the threshold at the fusion, the Receiver Operating Characteristic (ROC) in Fig. 21 was obtained. The ROC in Fig. 21 demonstrates the ability of NNs to solve DDF problems efficiently.

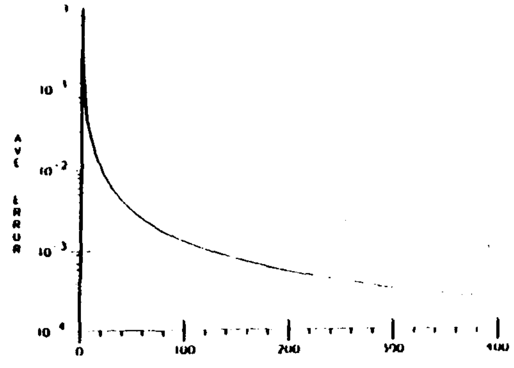


Fig. 20 Average fusion error per data set vs. epochs during training for different learning rates.
Solid line: Learning Rate = 0.9; Dotted line: Learning Rate = 0.5

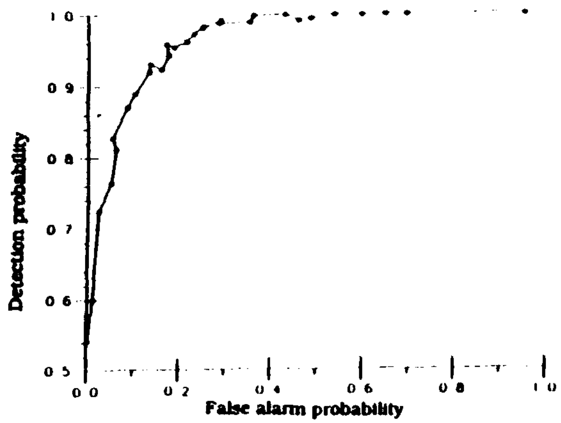


Fig. 21 Post-training Receiver Operating Characteristic (ROC) of NN DDF.

2. Training based on Neyman-Pearson

N-P training is conceptually identical to the backpropagation algorithm, except that training is done around a desired false alarm rate at the fusion. In order to achieve training around a desired false alarm rate α at the fusion, two possible performance criteria can be used to measure the output error:

$$E = P_M + \lambda(P_F - \alpha)^2 \tag{V.12}$$

or

$$E = P_M^2 + \lambda(P_F - \alpha)^2 \tag{V.13}$$

where P_M, P_F are the miss and false alarm probabilities at the fusion.

The modifications required to the standard backpropagation to implement the N-P fusion rule relate only to the energy function derivative with respect to the output. To get this, first we express the probabilities in term of the output as

$$P_M = \frac{\Sigma_{n=1}^{N}(1-u_0^n)u_1^n}{\Sigma_{n=1}^{N} u_1^n} \tag{V.14}$$

$$P_F = \frac{\Sigma_{n=1}^{N}(1-u_1^n)u_0^n}{\Sigma_{n=1}^{N}(1-u_1^n)} \tag{V.15}$$

which gives two possible derivative forms

$$\frac{dE}{du_0^m} = -\frac{u_1^m}{\Sigma_{m=1}^{N} u_1^n} + 2\lambda(P_F - \alpha)\frac{(1-u_1^m)}{\Sigma_{n=1}^{N}(1-u_1^n)} \tag{V.16}$$

$$\frac{dE}{du_0^m} = -2P_M\frac{u_1^m}{\Sigma_{m=1}^{N} u_1^n} + 2\lambda(P_F - \alpha)\frac{(1-u_1^m)}{\Sigma_{m=1}^{N} u_1^n} \tag{V.17}$$

for the second one. If we set

$$\delta_0 = \frac{dE}{du_0^m} u_0^m (1 - u_0^m) \qquad (V.18)$$

where o is the output neuron, from then on the backpropagation rule proceeds as described before. The *N-P trained* rule was successfully used in centralized and distributed DDF. *For numerical results and discussion* of the N-P training rule, see [22].

3. Training based on Kalman Filter

The problem of training a NN can be viewed as a Kalman Filtering problem [23], [24]. *If the ideal (unknown) weights and thresholds of the* NN are identified with the state x(n) of a Kalman Filter, then these *weights should be time-invariant, thus satisfy the plant equation*

$$x(n+1) = x(n) \qquad (V.19)$$

The unknown state *x(n) in the NN is observed via the nonlinear output* equation

$$d(n) = h(x(n)) + v(n) \qquad (V.20)$$

where the error made from not knowing the weights and thresholds precisely is modeled as zero mean, random error v(n) with covariance matrix $E[v(n)v(n)^T] = R(n)$, a positive definite matrix. The nonlinear function h( ) takes into account all the threshold nonlinearities at each neuron at every layer. From the nonlinear Kalman Filter theory, the state *x(n)* can be estimated using the Extended Kalman Filter (EKF) with equations

$$\hat{x}(n+1) = \hat{x}(n) + K(n)[d(n) - h(\hat{x}(n))] \qquad (V.20)$$

$$K(n) = P(n)H(n)[R(n) + H^T(n)P(n)H(n)]^{-1} \qquad (V.21)$$

$$P(n+1) = P(n) - K(n)H^T(n)P(n) \qquad (V.22)$$

where $H(n)_{ij}$ is the derivative of the output node i with respect to weight j, computed as in the backpropagation. Also d(n) is the desired vector output of the output neurons. For more details on the use of the EKF for training the NN to perform DDF see [22].

VI. Summary

The two major evidence processing theories, namely Bayesian and Dempster-Shafer's, are presented as applied to the problem *of Distributed* Decision or Evidence Fusion. Some of the fundamental results in Bayesian and Neyman-Pearson DDF are presented. *It is shown that a Generalized* Evidence Processing theory, which is *a generalization* of the Bayesian DDF using multi-level logic at the local processor, *can provide a framework* that allows comparison of the performance of the Bayesian and D-S DDFs under certain conditions. To that extend, a theorem is developed that shows that if the objective is to maximize the detection probability at the fusion for fixed *false alarm probability, the* Bayesian DDF outperforms the D-S DDF when multi-level logic is used locally, i.e. at the sensors. Natural structural similarities between the Bayesian DDF solution and neural networks are exploited. It is shown that NNs can learn to solve *the DDF efficiently,* even in the absence of explicit statistical information about the channel.

References

[1] Dempster, A. P., "A Generalization of Bayesian Inference," Journal of the Royal Statistical Society, Vol. 30, 1968, pp. 205-247.

[2] Van Tress, H. L., Detection, Estimations, and Modulation Theory, *Vol. 1, John Wiley & Sons*, New York, 1968.

[3] Shafer, G. A., A Mathematical Theory of Evidence, Princeton

University Press, 1976.

[4] Tenney, R. R. and Sandell, N.R., Jr., "Detection with Distributed Sensors," IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-17, July 1981, pp. 501-510.

[5] Tsitsiklis, J., and Athans, M., "On the Complexity of Distributed Decision Problem," IEEE Trans. on Automatic Control, AC-30, Vol. 5, May 1985, pp. 440-446.

[6] Sadjadi, F. A., "Hypothesis Testing in a Distributed Environment," IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-22, March 1986, pp. 134-137.

[7] Chair, Z. and Varshney, P.K., "Optimal Data Fusion in Multiple Sensor Detection Systems," IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-22, No. 1, January 1986, pp. 98-101.

[8] Srinivasan, R., "Distributed Radar Detection Theory," IEEE Proceedings, Vol. 133, Pt.F, No. 1, February 1986, pp. 55-60.

[9] Thomopoulos, S. C. A., Viswanathan, R. and Bougoulias, D. K., "Optimal Decision Fusion in Multiple Sensor Systems," IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-23, No. 5, Sept. 1987.

[10] Thomopoulos, S. C. A., Bougoulias, D.K., and Zhang, L., "Optimal and Suboptimal Distributed Decision Fusion," SPIE Proceedings, Vol. 931, Sensor Fusion (1988), pp. 26-30.

[11] Viswanathan, R., Thomopoulos, S. C. A., and Tumuluri, R., "Optimal Serial Distributed Decision Fusion," IEEE Transactions on Aerospace and Electronic Systems, Vol. 24, No. 4, pp. 366-376, July 1988.

[12] Viswanathan, R., Ansari, A., and Thomopoulos, S. C. A., Optimal Partitioning of Observations in Distributed Detection," Abstracts of Intern. Symposium on Information Theory, Kobe, Japan, June 19-24, 1988, p. 195.

[13] Thomopoulos, S. C. A., and Okello, N. N., "Distributed Detection with Mismatched Senors," SPIE '88 Symposium on Advances in Intelligent Robotic Systems, Boston, Nov. 1988.

[14] Thomopoulos, S. C. A., Viswanathan, R. and Bougoulias, D. K., "Optimal and Suboptimal Distributed Decision Fusion," IEEE Transactions on Aerospace and Electronic Systems, Vol. 25, No. 5, Sept. 1989.

[15] Thomopoulos, S. C. A., "Sensor Integration and Data Fusion," Special Issue on Sensor Integration and Data Fusion," International Journal of Robotics, Vol. 7, No. 3, 1990, pp. 337-372.

[16] Thomopoulos, S. C. A., "Generalized Evidence Processing Theory," SPIE '90, Symposium on Advances in Intelligent Robotic Systems, Boston, Nov. 5-9, 1990.

[17] Galuza, M., Performance Evaluation of Generalized Evidence Theory in Distributed Detection Problems, M.S. Thesis, Dept. of Electr. & Comp. Engr., Penn State Univ., May 1990.

[18] Tsitsiklis, J. N., "External Properties of Likelihood Ratio Quantizers," 1990 preprint.

[19] Longo, M., Lookabaugh, T. D., and Gray, R. m., "Quantization for Decentralized Hypothesis Testing under Communiation Constraints," IEEE Trans. on Info. Th., IT-36, No. 2, March 1990, pp. 241-255.

[20] McClleland, J. L., and Rumelhart, D. E., Parallel and Distributed Processing, MIT Press, Cambridge, MA, 1987.

[21] Nilsson, N. J., The Mathematical Foundation of Learning Machines, Morgan Kaufman Publishers, San Mateo, CA, 1990.

[22] Thomopoulos, S. C. A., Papadakis, I. N. M., and Saninoglou, H., "Centralized and Distributed Decision Fusion with Adaptive Networks," SPIE Conference on Sensor Fusion, Boston, Nov. 1991

[23] S. Singhal, and L. Wu, "Training Multilayer Perceptrons with the Extended Kalman Algorithm," in Advances in Neural Network Information Processing Systems 1, Ed. D. S. Touresky, Morgam Kaufmann Publ., Palo Alto, CA, 1989.

[24] D. W. Ruck, S. K. Rogers, and P. S. Maybeck, "Back Propagation: A Degenerate Kalman Filter?," preprint.

APPENDIX

A. Proof of Theorem 5

Let the set of observations at the N sensors be $r := (r_1, r_2, \ldots, r_N)$ with $r_i \in R^n$, $i \in S_{N}$, the set of the first positive N integers. The r_i's are assumed statistically independent, and not necessarily identically distributed, under either hypothesis. Assuming binary hypothesis testing, let $p_{ij}(r_i)$ designate the probability density function (pdf) or r_i under hypothesis j, where j = 0, 1. Each sensor partitions (quantizes) each own observation using an M-level nonlinearity, and passes the partitioned information to the fusion. Let $u := (u_1, u_2, \ldots, u_N)$ be the partitioned information at the fusion, with $u_i \in S_M$, the set of the first M positive integers. The test at the fusion which, for a given U, maximizes the detection probability for fixed false alarm probability, is the Neyman-Pearson (N-P) test [VaTr '68]. Assuming that any desired false alarm probability at the fusion can be achieved by a nonrandomized N-P test, we are interested in characterizing the M-level partitions that maximize the detection probability at the fusion over all possible M-level nonlinearities, thus prove Theorem 5. The proof of Theorem 5 in case of randomized N-P test at the fusion can be obtained along the same lines, using some additional technical arguments similar to [14].

In order to show that the optimal nonlinearities are based on the Likelihood Ratio Test (LRT), we establish two lemmas first.

Let u_0 designate the decision at the fusion. Then, $u_0 = 1$ implies that the fusion decision favors the hypothesis H_1, while $u_0 = 0$ implies the hypothesis H_0. Define the following probabilities

$$\alpha_0 = \Pr(u_0 = 1 \mid H_0) \tag{a.1}$$

$$\beta_0 = \Pr(u_0 = 1 \mid H_1) \tag{a.2}$$

$$P(i, j, m) = \Pr(u_i = j \mid H_m), \quad m = 0, 1 \tag{a.3}$$

and the indicator function

$$I(i, j) = \begin{cases} 1 & \text{if } u_i = j \\ 0 & \text{otherwise} \end{cases} \tag{a.4}$$

with $i \in S_N$ and $j \in S_M$. The LRT at the fusion is given as in Theorem 3, i.e.

$$\Lambda(u) = \frac{dP(u \mid H_1)}{dP(u \mid H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \lambda_0 \tag{a.5}$$

Due to the independence assumption, the LRT can be written in a product form as

$$\Lambda(u) = \frac{dP(u_1 \mid H_1) \ldots dP(u_N \mid H_1)}{dP(u_1 \mid H_0) \ldots dP(u_N \mid H_0)} = \prod_{i=1}^{N} \Lambda(u_i) \underset{H_0}{\overset{H_1}{\gtrless}} \lambda_0 \tag{a.6}$$

or, equivalently as

$$\sum_{i=1}^{N} I_i^T W_i \underset{H_0}{\overset{H_1}{\gtrless}} \lambda_0 \tag{a.7}$$

where, for $i \in S_N$,

$$I_i^T := [\, I(i,M), I(i,M-1), \ldots, I(i,1) \,] \tag{a.8}$$

$$W_i^T := [\, \ln \frac{dP(i,M,1)}{dP(i,M,0)}, \ \ln \frac{dP(i,M-1,1)}{dP(i,M-1,0)}, \ \ldots, \ \ln \frac{dP(i,1,1)}{dP(i,1,0)} \,]$$
(a.9)

As seen from (a.7), the actual value assigned to u_i's in (a.4) does not affect the decision at the fusion. What matters is that each u_i belongs to a set of cardinality M.

Assuming a nonrandomized test at the fusion, the LRT (a.6) partitions the set of all possible u's into two sets N_1 and N_2, such that N_1 is the set of all u sequences for which $\Lambda(u) \le \lambda_0$, and N_2 is the set of remaining sequences. The false alarm and detection probabilities can be expressed respectively as

$$\alpha_0 = \sum_{N_2} \prod_{i=1}^{N} I_i^T P(i,0) \tag{a.10}$$

$$\beta_0 = \sum_{N_2} \prod_{i=1}^{N} I_i^T P(i,1) \tag{a.11}$$

where $P(i,m) := [\, P(i,L,m), P(i,L-1,m), \ldots, P(i,1,m) \,]$ for $m = 0, 1$. We establish the following *Lemma 1*.

<u>***Lemma 1***</u>

Monotone Property:

Assume that

$$\frac{P(i,1,1)}{P(i,1,0)} \le \frac{P(i,2,1)}{P(i,2,0)} \le \ldots \le \frac{P(i,M,1)}{P(i,M,0)} \text{ for every } i \in S_N. \tag{a.12}$$

Let u_i assume some specific value u_i^0, $u_i^0 \in S_M$ for all i except ρ. Suppose that the set N_2 contains the sequence $(u_1^0, \ldots, u_{\rho-1}^0, u_\rho^0 = k, u_{\rho+1}^0, \ldots, u_N^0)$, for $k \in S_M$. Then, the sequences $(u_1^0, \ldots, u_{\rho-1}^0, u_\rho^0 = q, u_{\rho+1}^0, \ldots, u_N^0)$, $M \ge q \ge k+1$, are also contained in N_2.

<u>Proof</u>

The sequence $u = (u_1^0, \ldots, u_{\rho-1}^0, u_\rho^0 = k, u_{\rho+1}^0, \ldots, u_N^0)$ satisfies $\Lambda(u) > \lambda_0$. If the condition (a.12) is satisfied, equation (a.6) implies that each sequence $(u_1^0, \ldots, u_{\rho-1}^0, u_\rho^0 = q, u_{\rho+1}^0, \ldots, u_N^0)$, $M \geq q \geq k+1$, also satisfies $\Lambda(u) > \lambda_0$. □

The next lemma shows that quantization of the data based on the likelihood ratio satisfies the condition (a.12) in Lemma 1.

Lemma 2 ***Monotone property of the LRQ***

Let the quantization at the i-th sensor be

$$u_i = j \text{ if and only if } \lambda_{i,j} \leq \Lambda(r_i) < \lambda_{i,j+1} \qquad \text{(a.13)}$$

where $j \in S_M$, the $\lambda_{i,j}$'s are nondecreasing with $\lambda_{i,1} = 0$, $\lambda_{i,M+1} = \infty$, and $\Lambda(r_i)$ is the LR

$$\Lambda(r_i) = \frac{dP_1(r_i)}{dP_0(r_i)} \qquad \text{(a.14)}$$

For the quantizer (a.13), the inequalities in (a.12) are satisfied.

Proof

We prove the last inequality on the right hand side of (12). The other inequalities follow by similar reasoning. For the LRQ in (a.13) we have

$$P(i,j,m) = \int_{\lambda_{i,j}}^{\lambda_{i,j+1}} dP_\Lambda(\Lambda(r_i)|H_m) \quad m = 0, 1 \qquad \text{(a.15)}$$

where $dP_\Lambda(\Lambda(r_i)|H_m)$ is the probability distribution of the LR in (a.14) under hypothesis H_m. From the well-known fact [VTre '64] that

$$dP_\Lambda(\Lambda|H_1) = \Lambda\, dP_\Lambda(\Lambda|H_0) \qquad \text{(a.16)}$$

where the dependence of the LR on r_i has been omitted for notational brevity, it follows, by integrating both sides of (a.16) over $(\lambda_{i,j}, \lambda_{i,j+1})$ and lower bounding Λ by $\lambda_{i,j}$ in the right hand side integral, that

$$\frac{P(i,M,1)}{P(i,M,0)} \geq \lambda_{i,M} \qquad \text{(a.17)}$$

$$\frac{P(i,M-1,1)}{P(i,M-1,0)} \leq \lambda_{i,M} \qquad \text{(a.18)}$$

which proves the last inequality on the right hand side of (a.12). □

We are now in position to prove Theorem 5.

Proof of Theorem 5

Isolating the k-th sensor, equation (a.11) can be rewritten as

$$\beta_0 = \sum_{N_2} K_k \left[i_k^T P(k,1) \right] \qquad \text{(a.19)}$$

where

$$K_k = \prod_{\substack{i=1 \\ i \neq k}}^{N} i_i^T P(i,1) \qquad \text{(a.20)}$$

Since $\sum_{j=1}^{M} P(k,j,i) = 1$, without any loss of generality, $P(k,1,1)$ can be considered as function of the remaining $P(k,j,i)$'s for $j = 2, \ldots, M$. The expression in the square brackets of (a.19) can be written explicitly as

$$i_k^T P(k,1) = [\, l(k,M), \ldots, l(k,1) \,] \begin{bmatrix} P(k,M,1) \\ \vdots \\ P(k,1,1) \end{bmatrix} \qquad \text{(a.21)}$$

Let $(u_i = u_i^0$ for some $u_i^0 \in S_M$, $i \neq k$, $i \in S_N$, and $u_k = p$, $p \in S_M)$ be in the set N_2. Due to Lemma 1, the right hand side of (a.21) equals $\sum_{j=p}^{M} P(k,j,1)$. The existence of an optimum quantization implies that there exist probabilities $P(i,2,0)$, ..., $P(i,M,0)$, for every $i \in S_N$, such that the false alarm requirement (a.10) is satisfied. Hence, if there exists a quantization which attains the largest possible values for the probabilities $\{ \sum_{j=p}^{M} P(k,j,1),\ p = 2, \ldots, M,\ k \in S_N \}$, consistent with the false alarm requirement α_0, then such a quantization is optimum. Notice that the LRQ satisfies Lemma 1 (see Lemma 2). We complete the proof of the theorem by showing that the LRQ in (a.13) achieves the largest possible values for the probabilities $\{ \sum_{j=p}^{M} P(k,j,1),\ p = 2, \ldots, M,\ k \in S_N \}$.

Consider the LRQ T^* and any other quantization $\hat{T}$ such that

$$r_i \in C_\ell^* \text{ implies } u_i = \ell \text{ for } T^*,$$

$$r_i \in \hat{C}_\ell \text{ implies } u_i = \ell \text{ for } \hat{T},$$

$$\Pr(r_i \in C_\ell^* \mid H_m) = P^*(i,\ell,m), \tag{a.22}$$

$$\Pr(r_i \in \hat{C}_\ell \mid H_m) = \hat{P}(i,\ell,m),$$

and $\quad P^*(i,\ell,0) = \hat{P}(i,\ell,0)$.

In the above expressions, $\ell \in S_M$, $i \in S_N$, and C_ℓ^* $(\hat{C}_\ell)$ are mutually exclusive and collectively exhaustive subsets of R^n for every ℓ, i.e. they form a coverage of R^n. Denote the likelihood function $dP_m(r_i)$ under H_m as L_m, the integral $\int_{r_i \in R} dP_m(r_i)$ as $\int_R L_m$, the intersection of two sets S_1 and S_2 as $S_1 S_2$, and the compliment of a set S as $\bar{S}$.

Consider the difference

$$P^*(i,M,1) - \hat{P}(i,M,1) = \int_{C_M^*} L_1 - \int_{\hat{C}_M} L_1 \tag{a.23}$$

Upon adding and subtracting $\int_{C_M^* \hat{C}_M} L_1$ to the right hand side of (a.23).

$$P^*(i,M,1) - \hat{P}(i,M,1) = \int_{C_M^* \overline{\hat{C}_M}} L_1 - \int_{\hat{C}_M \overline{C_M}} L_1 \tag{a.24}$$

For the M-th quantization level threshold at the i-th sensor, $\lambda_{i,M} \le \frac{L_1}{L_0} < \infty$ holds in C_M^*, and $\frac{L_1}{L_0} < \lambda_{i,M}$ holds in $\overrightarrow{C_L}$. Hence, by making use of the (a.16), the right hand side of (a.24) is bounded from below by

$$\lambda_{i,M} \left[\int_{C_M^* \overline{\hat{C}_M}} L_0 - \int_{\hat{C}_M \overline{C_M}} L_0 \right].$$

Upon adding and subtracting $\int_{C_M^* \hat{C}_M} L_0$ to the above bound, we obtain

$$\overset{\bullet}{P}(l,M,1) - \hat{P}(l,M,1) \geq \lambda_{l,M}\left[\int_{\overset{\bullet}{C}_M} L_0 - \int_{\hat{C}_M} L_0 \right] \geq 0 \qquad \text{(a.25)}$$

where the last inequality follows from the requirement that $\overset{\bullet}{P}(l,M,0) = \hat{P}(l,M,0)$. Along similar lines, we could show that the following relations are true:

$$\sum_{j=p}^{M} \overset{\bullet}{P}(l,j,1) - \sum_{j=p}^{M} \hat{P}(l,j,1) = \int_{\bigcup_{k=p}^{M} \overset{\bullet}{C}_k} L_1 - \int_{\bigcup_{k=p}^{M} \hat{C}_k} L_1 \geq 0 \qquad \text{(a.26)}$$

for p = 2, 3, ..., M-1, where "U" stands for set union in the above. Furthermore, the inequalities in (a.25) and (a.26) are satisfied for every $l \in S_N$. This completes the proof of Theorem 5. □

INFORMATION SCIENCES 66, 91–118 (1992)

Distributed Decision Fusion in the Presence of Networking Delays and Channel Errors

STELIOS C. A. THOMOPOULOS*

Decision and Control Systems Laboratory, Department of Electrical and Computer Engineering, Pennsylvania State University, University Park, Pennsylvania 16802

and

LEI ZHANG

Department of Electrical Engineering, University of Maryland, College Park, Maryland 20740

ABSTRACT

The effects of transmission delay and channel errors on the performance of a distributed sensor system are studied. In a network of distributed sensors at a given time instant, the decisions from some sensors may not be available at the fusion center owing to networking and transmission delays. Assuming that the fusion center has to make a decision on the basis of the data from the rest of the sensors, provided that at least one peripheral decision has been received, it is shown that the optimal decision rule that maximizes the probability of detection for fixed probability of false alarm at the fusion center is the Neyman-Pearson test at the fusion center and the sensors as well. Furthermore, it is shown that, in the case of noisy channels, the decision made by each sensor depends on the reliability of the corresponding transmission channel. Moreover, the probability of false alarm at the fusion is restricted by the channel errors. For a given decision rule, the probability of any channel being in error must be kept at a certain level to achieve a desired probability of false alarm at the fusion. A suboptimal but computationally efficient algorithm is developed to solve for the sensor and fusion thresholds sequentially. Numerical results are provided to demonstrate the closeness of the solutions obtained by the suboptimal algorithm to the optimal solutions.

1. INTRODUCTION

Problems dealing with distributed decision fusion have been receiving much attention in recent years [1–11] and [14]. In [7], [10], [11], and [14] it

*This research is sponsored by the SDIO/IST and managed by the Office of Naval Research under Contract N00014-86-k-0515.

655 Avenue of the Americas, New York, NY 10010 0020-0255/92/$5.00

is proved that for the parallel and serial sensor topology the globally optimal solution to the fusion problem that maximizes the probability of detection for fixed probability of false alarm when sensors transmit independent binary decisions to the fusion center consists of a Neyman-Pearson (NP) test at the fusion and Likelihood Ratio (LR) tests at all sensors. In [7], the optimality of the N-P/L-R test for the parallel topology was derived using the Lagrange multipliers method. An independent proof of the optimality of the N-P test for the distributed fusion problem of independent decisions is given in [14]. In [11], a suboptimal sequential solution to the fusion problem in terms of a one-dimensional search algorithm was developed. In [7] and [11] it is assumed that the channels over which the decisions were communicated from the sensors to the fusion were errorless and that there were no delays associated with the transmission of the decisions, i.e., all the decisions were available to the fusion at the time of fusing. In this paper we consider the distributed decision fusion problem when decisions are transmitted with delays resulting from networking and bandwidth constraints over noisy channels that introduce errors in the sensor decisions as received by the fusion [12], [13].

Consider the distributed sensor fusion system shown in Figure 1. To capture the effect of delays in the reception of the peripheral decisions by the fusion, a time-slotted decision system is assumed. The time axis is subdivided into decision intervals. At the end of each decision interval, the fusion makes a decision based on the peripheral decisions that became

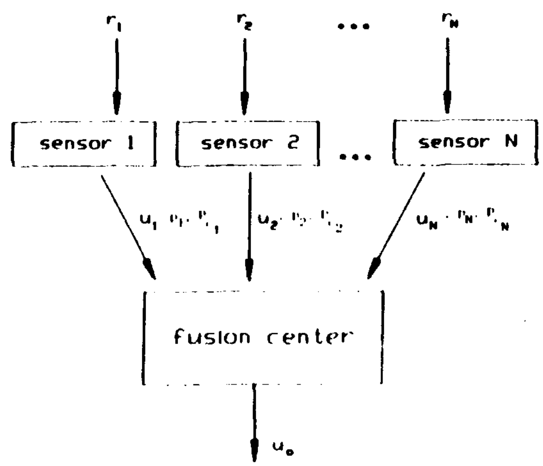


Fig. 1 Distributed fusion system.

available (arrived) during this decision interval and were transmitted by the sensors no earlier than one decision interval prior to the decision fusion time. Decisions that arrive at the fusion with delay greater than an interdecision time interval are simply ignored by the fusion. This requires that the decisions from the sensors are sent to the fusion time-marked. In between decisions, the fusion times out. We call this interval either *interdecision interval* or *time-out interval* for the fusion.

Let p_i $(i = 1, 2, \ldots, N)$ be the probability that u_i, the decision u_i of the ith peripheral sensor, is timely, i.e., no later than one interdecision interval from the time the local decision is reached until the time it is available for fusion at the fusion center. For the sake of analysis, it is assumed that the fusion makes decisions at the end of fixed time-length decision intervals, and times out between two consecutive decisions intervals. Then the p_i's correspond to the probability that the transmission delay does not exceed a time-out interval (= one decision interval). Hence, the delay model that is used for analysis in the paper does not depend on the delay distribution explicitely, but rather implicitly through the aggregate probabilities p_i. The p_i's can be thought of as the probability that the networking delay at the ith sensor does not exceed the time-out interval at the fusion. Let the binary decision u_i at the ith sensor takes on values

$$u_i = \begin{cases} 1, & \text{if the decision of the } i\text{th sensor favors hypothesis } H_1 \\ 0, & \text{if the decision of the } i\text{th sensor favors hypothesis } H_0 \end{cases}$$

The probability the decision u_i from the ith sensor is received correctly by the fusion is $1 - P_{e_i}$, where P_{e_i} is the probability that the noise in channel i has caused an error in u_i. The p_i's and P_{e_i}'s are the networking parameters and are assumed to be statistically independent of each other. The fusion makes a decision at time t on the basis of the data available at that time. The previous studied models in [7], [11], and [14] in which the decisions from the peripheral sensors are always available at the fusion center at the time of fusion, and are always received correctly, can be obtained as special cases of the models that are discussed in this paper by setting $p_1 = p_2 = \cdots = p_N = 1$, and $P_{e_1} = P_{e_2} = \cdots = P_{e_N} = 0$. Our objective is to investigate how the networking parameters affect the performance of the distributed decision fusion system and derive the optimal test that maximizes the probability of detection at the fusion for a fixed probability of false alarm.

This paper is organized in the following way. Section 2 contains the general definitions and notations used in later sections. In Section 3, 4, and 5, it is shown that the optimal test that maximizes the probability of detection at the fusion, for fixed probability of false alarm, is the Neyman-Pearson test at the sensors and the fusion in the case that either delay, or errors, or both are present. Furthermore, equations for the optimal set of thresholds analogous to those in [7] are obtained for all the three cases of delay, channel errors, or both delay and channel errors. In Section 6, a suboptimal sequential algorithm is derived that allows the determination of various near-to-the-optimal operating points. Numerical results are given in Section 7 and conclusions in Section 8.

2. DEFINITIONS AND NOTATIONS

For the model shown in Figure 1, the following notations are introduced to represent the decision sets received at the fusion center, the collections of such sets, and their probabilities of appearance at the fusion center:

$$U = \{u_{i_1}, u_{i_2}, \ldots, u_{i_m}\},\ 1 \leq m \leq N,\ i_1, \ldots, i_m \in \{1, 2, \ldots, N\}, \quad \text{and}$$

$$i_j \neq i_l \quad \text{for } j \neq l$$

is the set that contains at least one peripheral decision.

$$U^k = \{u_k, u_{i_1}, \ldots, u_{i_m}\},\ 0 \leq m \leq N-1,$$

$$i_1, \ldots, i_m \in \{1, \ldots, k-1, k+1, \ldots, N\}, \quad \text{and} \quad i_j \neq i_l, \quad \text{for } j \neq l$$

is the set that contains the decision from sensor k.

$$U_k = \{u_{i_1}, u_{i_2}, \ldots, u_{i_m}\},\ 1 \leq m \leq N,$$

$$i_1, \ldots, i_m \in \{1, 2, \ldots, k-1, k+1, \ldots N\}, \quad \text{and} \quad i_j \neq i_l, \quad \text{for } j \neq l$$

is the set that does not contain the decision from sensor k.

$$U_k^k = \{u_{i_1}, u_{i_2}, \ldots, u_{i_m}\},\ 0 \leq m \leq N-1,\ i_1, \ldots, i_m \in$$

$$\{1, k \quad 1, k+1, \ldots, N\}, \quad \text{and} \quad i_j \neq i_l, \quad \text{for } j \neq l$$

is the remaining set of U^k after removing u_k from U^k, i.e., $U_k^k \quad U^k \quad \{u_k\}$

S = the collection of all the sets that contain at least one peripheral decision

S^k = the collection of all the sets that contain u_k.

S_k = the collection of all the sets that do not contain u_k.

$S^{k,k \quad 1}$ = the collection of all the sets that contain both u_k and $u_{k \quad 1}$.

$S_k^{k \quad 1}$ = the collection of all the sets that contain $u_{k \quad 1}$ but not u_k.

$P(U)$ = Prob{set U received by fusion center at time t | at least one of the peripheral decisions has been received}.

3. OPTIMAL FUSION RULE WITH DELAYS AND IDEAL CHANNELS

Suppose that N sensors receive data from a common volume. Sensor k receives data r_k and generates the first stage binary decision u_k, $k = 1, 2, \ldots, N$. During a fusion interval the decisions are transmitted with random delays through errorless channels and arrive at the fusion center with some probability where they are combined into a final decision u_0 about which one of the two hypotheses is true where

$$u_j = \begin{cases} 1: & \text{decide hypothesis } H_1 \text{ is true} \\ 0: & \text{decide hypothesis } H_0 \text{ is true} \end{cases} \tag{1}$$

for $j = 0, 1, \ldots, N$.

We assume that the data r_k received by sensor k is statistically independent from the data received by the other sensors conditioned on each hypothesis. Hence, the decisions u_k are statistically independent conditioned on each hypothesis, $k = 1, 2, \ldots, N$. Given a desired level of probability of false alarm at the fusion center, $P_{F_0} = \alpha_0$, we seek the optimal test that maximizes the probability of detection P_{D_0} (or minimizes the probability of miss $P_{M_0} = 1 - P_{D_0}$). In our case, the P_{F_0} and P_{M_0} are given by

$$P_{F_0} = \sum_{U \in S} P_{F_0}(U)P(U) \tag{2}$$

and

$$P_{M_0} = \sum_{U \in S} P_{M_0}(U)P(U) \tag{3}$$

where

$$P_{F_0}(U) = \sum_{\{U\}} p(u_0 = 1|U)p(U|H_0), \tag{4}$$

$$P_{M_0}(U) = \sum_{\{U\}} p(u_0 = 0|U)p(U|H_1), \tag{5}$$

and

$$\sum_{U \in S} P(U) = 1. \tag{6}$$

Using an approach similar to [10], we define the Lagrangian

$$F = P_{M_0} + \lambda_0(P_{F_0} - \alpha_0) \tag{7}$$

Using equations (2), (3), (4), and (5) in [7], we obtain

$$F = \sum_{U \in S} \sum_{\{U\}} p(u_0 = 1|U)\left[\lambda_0 p(U|H_0) - p(U|H_1)\right]P(U) + 1 - \lambda_0\alpha_0 \tag{8}$$

Minimization of F at the fusion center is achieved if we choose the following decision rule:

$$d(U) = p(u_0 = 1|U)$$

$$= \begin{cases} 1 & \text{if } p(U|H_1) - \lambda_0 p(U|H_0) > 0\text{: decide } H_1 \\ 0 & \text{if } P(U|H_1) - \lambda_0 p(U|H_0) < 0\text{: decide } H_0 \end{cases} \tag{9}$$

or equivalently

$$\frac{p(U|H_1)}{p(U|H_0)} \underset{H_0 \atop u_0=0}{\overset{u_0=1 \atop H_1}{\gtrless}} \lambda_0 \tag{10}$$

where U could be any set that contains at least one peripheral decision.

Hence, the optimal fusion rule is an N-P test with threshold λ_0. The quantity F can be minimized further by minimizing the term inside the brackets in (8)

$$\lambda_0 p(U|H_0) - p(U|H_1) \tag{11}$$

with respect to the local thresholds.

Using the notations defined in Section 2, F can be expressed as

$$F = \sum_{U^k \in S^k} \sum_{\{U^k\}} d(U^k)\left[\lambda_0 p(U^k|H_0) - p(U^k|H_1)\right]P(U^k)$$

$$+ \sum_{U_k \in S_k} \sum_{\{U_k\}} d(U_k)\left[\lambda_0 p(U_k|H_0) - p(U_k|H_1)\right]P(U_k) + 1 - \lambda_0\alpha_0. \tag{12}$$

In (12), only the first term depends on the decision from sensor k, i.e., u_k, which can be expanded in terms of the decision rule at the kth sensor. Note that

$$p(u_k|H_i) = \int_{r_k} p(u_k|r_k)p(r_k|H_i); \qquad i = 0, 1 \tag{13}$$

we obtain that the first term in (12) is equal to:

$$\int_{r_k} p(u_k|r_k)\left[\lambda_0 C_0^k p(r_k|H_0) - C_1^k p(r_k|H_1)\right]$$

$$+ \sum_{U^k \in S^k} \sum_{\{U_k^k\}} d(u_k = 0, U_k^k)\left[\lambda_0 p(U_k^k|H_0) - p(U_k^k|H_1)\right]P(U^k) \tag{14}$$

where

$$C_i^k = \sum_{U^k \in S^k} \sum_{\{U_k^k\}} \left[d(u_k = 1, U_k^k) - d(u_k = 0, U_k^k) \right] p(U_k^k \mid H_i) P(U^k);$$

$$i = 0, 1. \quad (15)$$

By substituting (14) in (12), we obtain

$$F = \int_{r_k} p(u_k = 1 \mid r_k) \left[\lambda_0 C_0^k p(r_k \mid H_0) - C_1^k p(r_k \mid H_1) \right] + F_k \qquad (16)$$

where

$$F_k = \sum_{U^k \in S^k} \sum_{\{U_k^k\}} d(u_k = 0, U_k^k) \left[\lambda_0 p(U_k^k \mid H_0) - p(U_k^k \mid H_1) \right] P(U^k)$$
$$+ \sum_{U_k \in S_k} \sum_{\{U_k\}} d(U_k) \left[\lambda_0 p(U_k \mid H_0) - p(U_k \mid H_1) \right] P(U_k) + 1 - \lambda_0 \alpha_0. \qquad (17)$$

Thus, minimization of F w.r.t. the decision rule at the kth sensor is achieved, if the decision rule

$$p(u_k = 1 \mid r_k) = \begin{cases} 1 & \text{if } C_1^k p(r_k \mid H_1) - \lambda_0 C_0^k p(r_k \mid H_0) > 0\text{: decide } H_1 \\ 0 & \text{if } C_1^k p(r_k \mid H_1) - \lambda_0 C_0^k p(r_k \mid H_0) < 0\text{: decide } H_0 \end{cases} \qquad (18)$$

is chosen, or equivalently

$$\frac{p(r_k \mid H_1)}{p(r_k \mid H_0)} \underset{u_k = 0}{\overset{u_k = 1}{\gtrless}} \lambda_0 \frac{C_0^k}{C_1^k}. \qquad (19)$$

Hence, the thresholds at the peripheral sensors are given by

$$\lambda_k = \lambda_0 \frac{C_0^k}{C_1^k}, \qquad k = 1, 2, \ldots, N. \qquad (20)$$

Equation (20) leads to the conclusion that the optimal solution of the problem involves an N-P test at each sensor and the fusion center as well.

4. OPTIMAL FUSION RULE IN NOISY CHANNELS

Next we consider the case in which the transmission delays are zero ($p_i = 1$, for $i = 1, 2, \ldots, N$) but the channels are noisy. In this case, the decision from each sensor is always available at the fusion center, i.e., the set U received by fusion always contains N peripheral decisions. Let P_{c_i} be the probability that the decision u_i from the ith sensor is, $i = 1, 2, \ldots, N$. The probability of false alarm P_{f_0} and the probability of miss P_{M_0} at the fusion center are given by

$$P_{f_0} = \sum_{\{U\}} p(u_0 = 1 \mid U) \bar{p}(U \mid H_0) \qquad (21)$$

and

$$P_{M_0} = \sum_{\{U\}} p(u_0 = 0 \mid U) \bar{p}(U \mid H_1) \qquad (22)$$

where

$$\bar{p}(U \mid H_i) = \prod_{k=1}^{N} \bar{p}(u_k \mid H_i)$$
$$= \prod_{k=1}^{N} \left[p(u_k \mid H_i, c_k)(1 - P_{c_k}) + p(u_k \mid H_i, \bar{c}_k) P_{c_k} \right]$$
$$= \prod_{k=1}^{N} \left[p(u_k \mid H_i)(1 - P_{c_k}) + (1 - p(u_k \mid H_i)) P_{c_k} \right] \qquad (23)$$

and

$$c_k = \{\text{The decision } u_k \text{ is received correctly}\}, \tag{24}$$

$$\bar{c}_k = \{\text{The decision } u_k \text{ is received incorrectly}\}, \tag{25}$$

$$p(u_k \mid H_i) = p(u_k \mid H_i, c_k). \tag{26}$$

Define the Lagrangian

$$F = P_{M_0} + \lambda_0 (P_{F_0} - \alpha_0)$$

$$= \sum_{\{U\}} p(u_0 = 1 \mid U) \left[\lambda_0 \bar{p}(U \mid H_0) - \bar{p}(U \mid H_1) \right] + 1 - \lambda_0 \alpha_0. \tag{27}$$

The minimum value of F at the fusion then is achieved if the decision rule

$$d(U) = p(u_0 = 1 \mid U)$$

$$= \begin{cases} 1 & \text{if } \bar{p}(U \mid H_1) - \lambda_0 \bar{p}(U \mid H_0) > 0\text{: decide } H_1 \\ 0 & \text{if } \bar{p}(U \mid H_1) - \lambda_0 \bar{p}(U \mid H_0) < 0\text{: decide } H_0 \end{cases} \tag{28}$$

is chosen, or equivalently

$$\frac{\bar{p}(U \mid H_1)}{\bar{p}(U \mid H_0)} \underset{\substack{H_0 \\ u_0 = 0}}{\overset{\substack{u_0 = 1 \\ H_1}}{\gtrless}} \lambda_0. \tag{29}$$

Since

$$p(u_k \mid H_i) = \int_{r_k} \left[p(u_k \mid r_k)(1 - P_{c_k}) + (1 - p(u_k \mid r_k)) P_{c_k} \right] p(r_k \mid H_i) \tag{30}$$

where

$$p(u_k \mid r_k) = p(u_k \mid r_k, c_k), \tag{31}$$

the Lagrangian in (27) can be expanded as

$$F = \int_{r_k} p(u_k = 1 \mid r_k) \left[(1 - P_{c_k}) - P_{c_k} \right]$$

$$\times \left[\lambda_0 C_0^k p(r_k \mid H_0) - C_1^k p(r_k \mid H_1) \right] + F_k, \tag{32}$$

where

$$C_i^k = \sum_{\{U_k\}} \left[d(u_k = 1, U_k) - d(u_k = 0, U_k) \right] \bar{p}(U_k \mid H_i), \tag{33}$$

$$F_k = P_{c_k} \left[\lambda_0 C_0^k - C_1^k \right] + \sum_{\{U_k\}} d(u_k = 0, U_k)$$

$$\times \left[\lambda_0 \bar{p}(U_k \mid H_0) - \bar{p}(U_k \mid H_1) \right] + 1 - \lambda_0 \alpha_0. \tag{34}$$

From (32) it follows that F is minimized w.r.t. the decision rule of the kth sensor if

Case 1. $(1 - P_{c_k}) - P_{c_k} > 0$, or $P_{c_k} < 0.5$ and the decision rule

$$p(u_k = 1 \mid r_k) = \begin{cases} 1 & \text{if } C_1^k p(r_k \mid H_1) - \lambda_0 C_0^k p(r_k \mid H_0) > 0\text{: decide } H_1 \\ 0 & \text{if } C_1^k p(r_k \mid H_1) - \lambda_0 C_0^k p(r_k \mid H_0) < 0\text{: decide } H_0 \end{cases} \tag{35}$$

is chosen, or equivalently

$$\frac{p(r_k \mid H_1)}{p(r_k \mid H_0)} \underset{u_k = 0}{\overset{u_k = 1}{\gtrless}} \lambda_0 \frac{C_0^k}{C_1^k} = \lambda_k; \qquad k = 1, 2, \ldots, N. \tag{36}$$

Case 2. $(1 - P_{c_k}) - P_{c_k} < 0$, or $P_{c_k} > 0.5$ and the decision rule

$$p(u_k = 1 | r_k) = \begin{cases} 0 & \text{if } C_1^k p(r_k | H_1) - \lambda_0 C_0^k p(r_k | H_0) > 0\text{: decide } H_1 \\ 1 & \text{if } C_1^k p(r_k | H_1) - \lambda_0 C_0^k p(r_k | H_0) < 0\text{: decide } H_0 \end{cases} \tag{37}$$

is chosen, or equivalenty

$$\frac{p(r_k | H_1)}{p(r_k | H_0)} \underset{u_k = 0}{\overset{u_k = 1}{\lessgtr}} \lambda_0 \frac{C_0^k}{C_1^k} = \lambda_k; \qquad k = 1, 2, \ldots, N. \tag{38}$$

Equations (35) through (38) indicate that the optimal decision rule at the sensor is dependent on the corresponding channel error. There exists a switching point $P_{c_k} = 0.5$ such that, if the probability of channel error is less than 0.5 the sensor will transmit the true decision to the fusion; otherwise it will transmit its compliment.

5. OPTIMAL FUSION RULE WITH DELAY AND NOISY CHANNELS

When both transmission delay and channel error are considered, the optimal decision rule at the fusion center is given by

$$\frac{\bar{p}(U | H_1)}{\bar{p}(U | H_0)} \underset{\substack{H_0 \\ u_0 = 0}}{\overset{\substack{u_0 = 1 \\ H_1}}{\gtrless}} \lambda_0 \tag{39}$$

where

$$\bar{p}(U | H_i) = \prod_{u_k \in U} \left[p(u_k | H_i)(1 - P_{c_k}) + (1 - p(u_k | H_i)) P_{c_k} \right] \tag{40}$$

and

$$c_k := \{\text{The decision } u_k \text{ is received correctly}\}, \tag{41}$$

$$p(u_k | H_i) = p(u_k | H_i, c_k). \tag{42}$$

The local decision rule is as follows:
If $P_{c_k} < 0.5$, then

$$\frac{p(r_k | H_1)}{p(r_k | H_0)} \underset{u_k = 0}{\overset{u_k = 1}{\gtrless}} \lambda_0 \frac{D_0^k}{D_1^k} = \lambda_k; \qquad k = 1, 2, \ldots, N. \tag{43}$$

If $P_{c_k} > 0.5$, then

$$\frac{p(r_k | H_1)}{p(r_k | H_0)} \underset{u_k = 0}{\overset{u_k = 1}{\lessgtr}} \lambda_0 \frac{D_0^k}{D_1^k} = \lambda_k; \qquad k = 1, 2, \ldots, N. \tag{44}$$

where, for $i = 0, 1$,

$$D_i^k = \sum_{U^k \in S^k} \sum_{\{U_k^k\}} \left[d(u_k = 1, U_k^k) - d(u_k = 0, U_k^k) \right] \bar{p}(U_k^k | H_i) P(U^k) \tag{45}$$

6. SUBOPTIMAL SOLUTION

In Equations (20), (36), and (43) the optimal set of thresholds is given in terms of a set of nonlinear coupled equations whose solution depends on the fusion policy, which is unknown. Hence, with the exception of very few simple cases, the equations that determine the optimal thresholds as obtained by the Lagrangian approach cannot be solved. Hence, a multidimensional search over all possible operating points of the sensors and all fusion policies needs to be carried through, if the optimal solution is sought. This, however, can be computationally tedious or even infeasible [12–13].

Hence, a computationally efficient suboptimal algorithm has been developed to solve for the thresholds sequentially by minimizing the residual terms in the Lagrangian instead of the entire F [11]. This is equivalent to assuming that for each sensor whose threshold is being determined, the operating points of the sensors whose thresholds were determined in previous steps are set at the extreme points $P_F = P_D = 0$.

First consider the case of errorless channels. Choose an arbitrary sensor k, and minimize F w.r.t. this sensor first. The optimal threshold for this sensor is given by (19) and the residual term of F, i.e., F_k, takes the form of (17). F can be minimized further by minimizing F_k w.r.t. another sensor, say sensor $k-1$. Expand F_k with respect to the decision rule of the $(k-1)$th sensor and we obtain

$$F_k = \sum_{U^{k,k-1}\in S^{k,k-1}} \sum_{\{U_k^{k,k-1}\}} d(u_k = 0, U_k^{k,k-1})$$

$$\times\left[\lambda_0 p(U_k^{k,k-1}\,|\,H_0) - p(U_k^{k,k-1}\,|\,H_1)\right] P(U^{k,k-1})$$

$$+ \sum_{U_{k-1}^k \in S_k^k} \sum_{\{U_{k,k-1}^k\}} d(u_k = 0, U_{k,k-1}^k)$$

$$\times\left[\lambda_0 p(U_{k,k-1}^k\,|\,H_0) - p(U_{k,k-1}^k\,|\,H_1)\right] P(U_{k-1}^k)$$

$$+ \sum_{U_k^{k-1}\in S_k^{k-1}} \sum_{\{U_k^{k-1}\}} d(U_k^{k-1})$$

$$\times\left[\lambda_0 p(U_k^{k-1}\,|\,H_0) - p(U_k^{k-1}\,|\,H_1)\right] P(U_k^{k-1})$$

$$+ \sum_{U_{k,k-1}\in S_{k,k-1}} \sum_{\{U_{k,k-1}\}} d(U_{k,k-1})$$

$$\times\left[\lambda_0 p(U_{k,k-1}\,|\,H_0) - p(U_{k,k-1}\,|\,H_1)\right] P(U_{k,k-1}) + 1 - \lambda_0\alpha_0. \quad (46)$$

In (46), only the first and the third terms depends on the decision from the $(k-1)$th sensor, and they can be expressed as:

$$\text{1st term in (46)} = \int_{r_{k-1}} p(u_{k-1} = 1\,|\,r_{k-1})$$

$$\times\left[\lambda_0 C_0^{k,k-1} p(r_{k-1}\,|\,H_0) - C_1^{k,k-1} p(r_{k-1}\,|\,H_1)\right]$$

$$+ \sum_{U^{k,k-1}\in S^{k,k-1}} \sum_{\{U^{k,k-1}\}} d(u_k = 0, u_{k-1} = 0, U_{k,k-1}^{k,k-1})$$

$$\times\left[\lambda_0 p(U_{k,k-1}^{k,k-1}\,|\,H_0) - p(U_{k,k-1}^{k,k-1}\,|\,H_1)\right] P(U^{k,k-1}) \quad (47)$$

$$\text{3rd term in (46)} = \int_{r_{k-1}} p(u_{k-1} = 1\,|\,r_{k-1})$$

$$\times\left[\lambda_0 C_0^{k-1} p(r_{k-1}\,|\,H_0) - C_1^{k-1} p(r_{k-1}\,|\,H_1)\right]$$

$$+ \sum_{U_k^{k-1}\in S_k^{k-1}} \sum_{\{U_{k,k-1}^{k-1}\}} d(u_{k-1} = 0, U_{k,k-1}^{k-1})$$

$$\times\left[\lambda_0 p(U_{k,k-1}^{k-1}\,|\,H_0) - p(U_{k,k-1}^{k-1}\,|\,H_1)\right] P(U_k^{k-1}) \quad (48)$$

where, for $i = 0, 1$,

$$C_i^{k,k-1} = \sum_{U^{k,k-1}\in S^{k,k-1}} \sum_{\{U_{k,k-1}^{k,k-1}\}}$$

$$\left[d(u_k = 0, u_{k-1} = 1, U_{k,k-1}^{k,k-1}) - d(u_k = 0, u_{k-1} = 0, U_{k,k-1}^{k,k-1})\right]$$

$$\times p(U_{k,k-1}^{k,k-1}\,|\,H_i) P(U^{k,k-1}), \quad (49)$$

$$C_i^{k-1} = \sum_{U_k^{k-1}\in S_k^{k-1}} \sum_{\{U_{k,k-1}^{k-1}\}} \left[d(u_{k-1} = 1, U_{k,k-1}^{k-1}) - d(u_{k-1} = 0, U_{k,k-1}^{k-1})\right]$$

$$\times p(U_{k,k-1}^{k-1}\,|\,H_i) P(U_k^{k-1}). \quad (50)$$

Using (47) and (48) in (46), we obtain

$$F_k = \int_{r_{k-1}} p(u_{k-1} = 1\,|\,r_{k-1})\left[\lambda_0 (C_0^{k,k-1} + C_0^{k-1}) p(r_{k-1}\,|\,H_0)\right.$$

$$\left. - (C_1^{k,k-1} + C_1^{k-1}) p(r_{k-1}\,|\,H_1)\right] + F_{k,k-1} \quad (51)$$

where the residual term $F_{k,k-1}$ is given by

$$
\begin{aligned}
F_{k,k-1} = & \sum_{U^{k,k-1} \in S^{k,k-1}} \sum_{(U^{k,k}_{k,k-1})} d(0,0,U^{k,k}_{k,k-1}) \\
& \times \left[\lambda_0 p(U^{k,k}_{k,k-1} \mid H_0) - p(U^{k,k}_{k,k-1} \mid H_1) \right] P(U^{k,k-1}) \\
& + \sum_{U^k_{k-1} \in S^k_{k-1}} \sum_{(U^k_{k,k-1})} d(0,U^k_{k,k-1}) \\
& \times \left[\lambda_0 p(U^k_{k,k-1} \mid H_0) - p(U^k_{k,k-1} \mid H_1) \right] P(U^k_{k-1}) \\
& + \sum_{U^{k-1}_k \in S^{k-1}_k} \sum_{(U^{k-1}_{k,k-1})} d(0,U^{k-1}_{k,k-1}) \\
& \times \left[\lambda_0 p(U^{k-1}_{k,k-1} \mid H_0) - p(U^{k-1}_{k,k-1} \mid H_1) \right] P(U^{k-1}_k) \\
& + \sum_{U_{k,k-1} \in S_{k,k-1}} \sum_{(U_{k,k-1})} d(U_{k,k-1}) \\
& \times \left[\lambda_0 p(U_{k,k-1} \mid H_0) - p(U_{k,k-1} \mid H_1) \right] P(U_{k,k-1}) + 1 - \lambda_0 \alpha_0 .
\end{aligned}
\tag{52}
$$

Furthermore, minimization of F_k in (51) with respect to the decision rule at the $(k-1)$th sensor, results in the following (suboptimal) decision rule for the $k-1$ peripheral sensors:

$$
p(u_{k-1} = 1 \mid r_{k-1}) = \begin{cases} 1 & \text{if } (C_1^{k,k-1} + C_1^{k-1}) p(r_{k-1} \mid H_1) \\ & \quad - \lambda_0 (C_0^{k,k-1} + C_0^{k-1}) p(r_{k-1} \mid H_0) > 0 \colon H_1 \\ 0 & \text{if } (C_1^{k,k-1} + C_1^{k-1}) p(r_{k-1} \mid H_1) \\ & \quad - \lambda_0 (C_0^{k,k-1} + C_0^{k-1}) p(r_{k-1} \mid H_0) < 0 \colon H_0 \end{cases}
\tag{53}
$$

or,

$$
\frac{p(r_{k-1} \mid H_1)}{p(r_{k-1} \mid H_0)} \underset{u_{k-1}=0}{\overset{u_{k-1}=1}{\gtrless}} \lambda_0 \frac{C_0^{k,k-1} + C_0^{k-1}}{C_1^{k,k-1} + C_1^{k-1}} := \lambda_{k-1}.
\tag{54}
$$

Following the same procedure as that just shown, the residual term $F_{k,k-1}$ can be minimized further w.r.t. the $k-2$ sensor. Minimization yields the following threshold test for the $k-2$ sensor:

$$
\frac{p(r_{k-2} \mid H_1)}{p(r_{k-2} \mid H_0)} \underset{u_{k-2}=0}{\overset{u_{k-2}=1}{\gtrless}} \lambda_0 \frac{C_0^{k,k-1,k-2} + C_0^{k,k-2} + C_0^{k-1,k-2} + C_0^{k-2}}{C_1^{k,k-1,k-2} + C_1^{k,k-2} + C_1^{k-1,k-2} + C_1^{k-2}} = \lambda_{k-2}.
\tag{55}
$$

By repeating the procedure, the thresholds for all N sensors can be obtained iteratively. For the last sensor, i.e., sensor 1, the threshold is equal to the threshold at the fusion center and the decision rule

$$
\frac{p(r_1 \mid H_1)}{p(r_1 \mid H_0)} \underset{u_1=0}{\overset{u_1=1}{\gtrless}} \lambda_0 = \lambda_1.
\tag{56}
$$

For the case in which both delay and channel error are considered, similar results can be obtained by applying the same procedure as that just shown. In that case, the suboptimal decision rule for this case at the kth sensor is given by the following:

If $P_{e_k} < 0.5$, then

$$
\frac{p(r_k \mid H_1)}{p(r_k \mid H_0)} \underset{u_k=0}{\overset{u_k=1}{\gtrless}} \lambda_0 \frac{D_0^k}{D_1^k} = \lambda_k .
\tag{57}
$$

If $P_{e_k} > 0.5$, then

$$
\frac{p(r_k \mid H_1)}{p(r_k \mid H_0)} \underset{u_k=0}{\overset{u_k=1}{\lessgtr}} \lambda_0 \frac{D_0^k}{D_1^k} = \lambda_k .
\tag{58}
$$

where

$$D_i^k = \sum_{U^k \in S^k} \sum_{(U_k^k)} \left[d(u_k = 1, U_k^k) - d(u_k = 0, U_k^k) \right] \bar{p}(U_k^k \mid H_i) P(U^k). \quad (59)$$

For the $k-1$ sensor, the decision rule becomes:
If $P_{e_{k-1}} < 0.5$, then

$$\frac{p(r_{k-1} \mid H_1)}{p(r_{k-1} \mid H_0)} \underset{u_{k-1}=0}{\overset{u_{k-1}=1}{\gtrless}} \lambda_0 \frac{D_0^{k,k-1} + D_0^{k-1}}{D_1^{k,k-1} + D_1^{k-1}} = \lambda_{k-1}. \quad (60)$$

If $P_{e_{k-1}} > 0.5$, then

$$\frac{p(r_{k-1} \mid H_1)}{p(r_{k-1} \mid H_0)} \underset{u_{k-1}=0}{\overset{u_{k-1}=1}{\lessgtr}} \lambda_0 \frac{D_0^{k,k-1} + D_0^{k-1}}{D_1^{k,k-1} + D_1^{k-1}} = \lambda_{k-1}. \quad (61)$$

where, for $i = 0, 1$,

$$D_i^{k,k-1} = \sum_{U^{k,k-1} \in S^{k,k-1}} \sum_{(U_{k,k-1}^{k,k-1})} \left[d^k(u_k = 0, u_{k-1} = 1, U_{k,k-1}^{k,k-1}) \right.$$

$$\left. - d^k(u_k = 0, u_{k-1} = 0, U_{k,k-1}^{k,k-1}) \right].$$

$$\bar{p}(U_{k,k-1}^{k,k-1} \mid H_i) P(U^{k,k-1}), \quad (62)$$

$$D_i^{k-1} = \sum_{U_k^{k-1} \in S_k^{k-1}} \sum_{(U_{k,k-1}^{k-1})}$$

$$\times \left[d(u_{k-1} = 1, U_{k,k-1}^{k-1}) - d(u_{k-1} = 0, U_{k,k-1}^{k-1}) \right]$$

$$\times \bar{p}(U_{k,k-1}^{k-1} \mid H_i) P(U_k^{k-1}), \quad (63)$$

and

$$d^k(U_k^{k,k-1}) = P_{e_k} d(u_k = 1, U_k^{k,k-1}) + (1 - P_{e_k}) d(u_k = 0, U_k^{k,k-1}) \quad (64)$$

is the average decision rule over the error probability of the kth sensor channel.

For the $k-2$ sensor, the (suboptimal) threshold is given by:
If $P_{e_k} < 0.5$, then

$$\frac{p(r_{k-2} \mid H_1)}{p(r_{k-2} \mid H_0)} \underset{u_{k-2}=0}{\overset{u_{k-2}=1}{\gtrless}} \lambda_0 \frac{D_0^{k,k-1,k-2} + D_0^{k,k-2} + D_0^{k-1,k-2} + D_0^{k-2}}{D_1^{k,k-1,k-2} + D_1^{k,k-2} + D_1^{k-1,k-2} + D_1^{k-2}} = \lambda_{k-2}. \quad (65)$$

If $P_{e_k} > 0.5$, then

$$\frac{p(r_{k-2} \mid H_1)}{p(r_{k-2} \mid H_0)} \underset{u_{k-2}=0}{\overset{u_{k-2}=1}{\lessgtr}} \lambda_0 \frac{D_0^{k,k-1,k-2} + D_0^{k,k-2} + D_0^{k-1,k-2} + D_0^{k-2}}{D_1^{k,k-1,k-2} + D_1^{k,k-2} + D_1^{k-1,k-2} + D_1^{k-2}} = \lambda_{k-2}. \quad (66)$$

where

$$d^{k,k-1}(U_{k,k-1}^{k,k-1,k-2}) = P_{e_{k-1}} d^k(u_{k-1} = 1, U_{k,k-1}^{k,k-1,k-2})$$

$$+ (1 - P_{e_{k-1}}) d^k(u_{k-1} = 0, U_{k,k-1}^{k,k-1,k-2}). \quad (67)$$

Finally, for the last sensor, i.e., sensor 1, the (suboptimal) decision rule becomes:
If $P_{e_1} < 0.5$, then

$$\frac{p(r_1 \mid H_1)}{p(r_1 \mid H_0)} \underset{u_1=0}{\overset{u_1=1}{\gtrless}} \lambda_0 = \lambda_1. \quad (68)$$

If $P_{e_1} > 0.5$, then

$$\frac{p(r_1 \mid H_1)}{p(r_1 \mid H_0)} \underset{u_1=0}{\overset{u_1=1}{\lessgtr}} \lambda_0 = \lambda_1. \quad (69)$$

7. NUMERICAL RESULTS

Performance evaluation of the distributed fusion system of Figure 1 in a slow fading Rayleigh channel has been done numerically for the case of three sensors ($N = 3$). For numerical convenience and for the sake of clarity of presentation, it was assumed that all of three sensors incurred *the same networking delays*, i.e., *that during a fusion interval*, they transmit their decisions to the fusion center with the same probability p, i.e., $p_1 = p_2 = p_3 = p$. Furthermore, and for the same reasons, it was assumed that the error probabilities in the different channels over which the sensors transmit their decision to the fusion were the same. Under these assumptions, the probabilities of detection and false alarm at the kth sensor are given by ([12] and [15]):

$$P_{F_k} = [\lambda_k(1+\epsilon_k)]^{-(1/\epsilon_k)} \tag{70}$$

$$P_{D_k} = P_{F_k}^{1/1+\epsilon_k} \tag{71}$$

where λ_k denotes the threshold of the kth sensor and ϵ_k the signal-to-noise ratio at the kth sensor.

In the case of $N = 3$, we have seven possible decision sets that may be received by the fusion center:

$$\begin{aligned} &U^1_{2,3} = \{u_1\} \qquad U^{1,2}_3 = \{u_1, u_2\} \qquad U^{1,2,3} = \{u_1, u_2, u_3\} \\ &U^2_{1,3} = \{u_2\} \qquad U^{2,3}_1 = \{u_2, u_3\} \\ &U^3_{1,2} = \{u_3\} \qquad U^{1,3}_2 = \{u_1, u_3\} \end{aligned} \tag{72}$$

with the following probabilities:

$$P(U^1_{2,3}) = P(U^2_{1,3}) = P(U^3_{1,2}) = \frac{p(1-p)^2}{1-(1-p)^3}, \tag{73}$$

$$P(U^{1,2}_3) = P(U^{2,3}_1) = P(U^{1,3}_2) = \frac{p^2(1-p)}{1-(1-p)^3}, \tag{74}$$

$$P(U^{1,2,3}) = \frac{p^3}{1-(1-p)^3}. \tag{75}$$

The fusion center can choose different combining policies according to the number of peripheral decisions received. For example, if the set U contains only one decision, the fusion center must choose this decision as

true (SURE rule), since there is no other choice. If the set U contains two decisions, the fusion can choose either an AND or OR policy to combine the decisions. If U contains three decisions, the majority logic (ML) can also be used in addition to AND and OR. Simulation results from various fusion rules in slow fading Rayleigh channels [12–13] are given next for different delays and channel error probabilities.

Figures 2 through 8 correspond to the ideal (errorless) channel cases. In Figures 2, 3, and 4 the probability of detection at the fusion for both optimal and suboptimal cases with equal signal-to-noise ratios (SNR's) is plotted as a function of the sensor's SNR and the networking (delay) parameter p under three different combining policies. The probability of false alarm is fixed at 10^{-6}. It can be seen that the P_{D_0} is a monotonic increasing function of p for both OR-OR and ML-OR policy. However, this is not true if the decision rule AND-AND is chosen. It is worth noticing the "folding effect" in the case of the AND-AND fusion rule, meaning that higher delay may yield higher probability of detection for the same probability of false alarm.

A comparison of three decision rules (OR, AND and ML-OR) for $p = 0.9$ is given in Figure 5. It is seen that the OR policy yields the best performance.

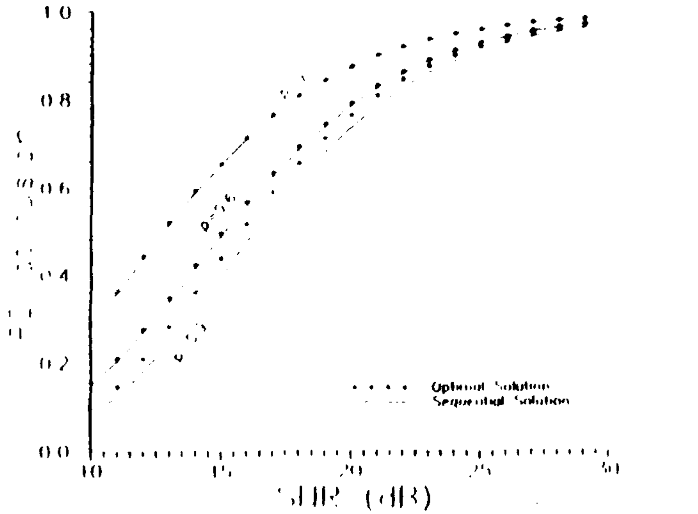


Fig. 2. Comparison of two solutions for $N = 3$ and equal SNR case under AND policy

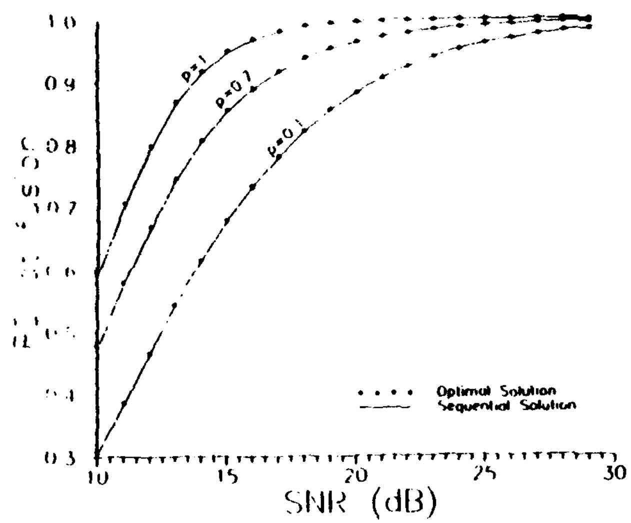


Fig. 3. Comparison of two solutions for $N = 3$ and equal SNR case under OR policy.

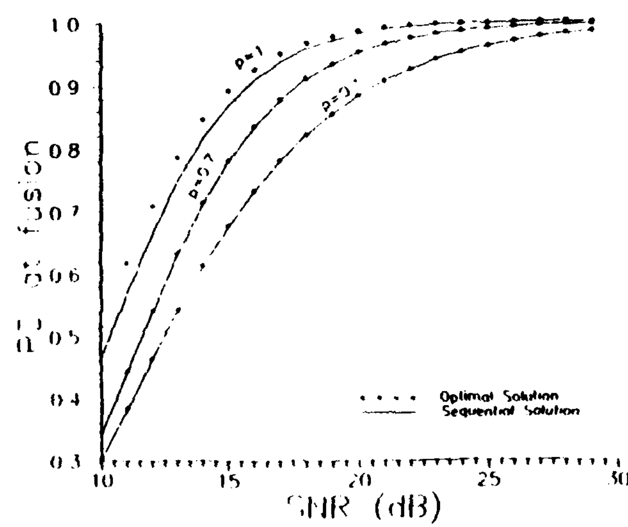


Fig. 4. Comparison of two solutions for $N = 3$ and equal SNR case under ML-OR policy.

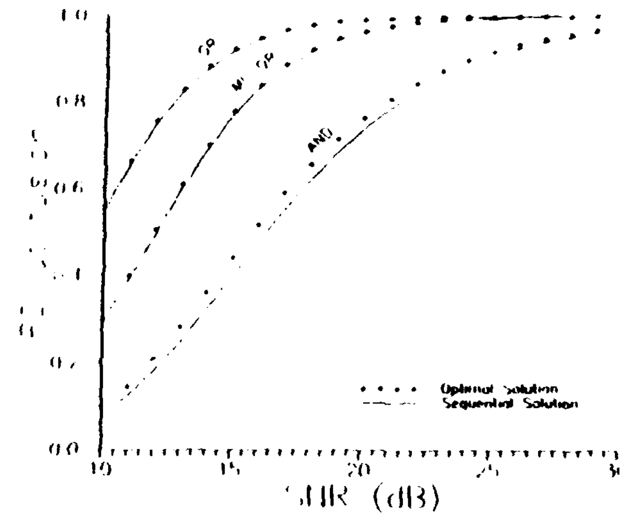


Fig. 5. Comparison of three policies for $N = 3$, $p = 0.9$, and equal SNR case.

In Figures 6 and 7 the thresholds of the sensors and the fusion center are plotted for $p = 1.0$ and 0.9, respectively, under the OR policy. Both optimal and sequential solutions are given for fixed $P_{F_0} = 10^{-6}$. The results show that the sensors operate at different thresholds in the sequential case.

Figure 8 gives the solutions for an unequal SNR case with $p = 0.9$ and $P_{F_0} = 10^{-6}$ under OR policy.

For all the cases just discussed, the sequential algorithm yields results that are very close to the optimal ones.

Figures 9 through 11 correspond to the noisy channel cases. The probability of false alarm at the fusion is fixed at 10^{-4}. For each policy, two sets of curves are plotted: one set without delay and another with delay. We assume that all three channels have the same probability of being in error, i.e., $P_{e_1} = P_{e_2} = P_{e_3} = P_e$. Computer simulation results have shown that if the OR policy is chosen, P_e must be kept at 10^{-5} level to achieve the desired $P_{F_0} = 10^{-4}$. However, for the AND policy, the same P_{F_0} can be achieved when P_e is at 10^{-3}.

8. CONCLUSIONS

The effect of transmission delays and errors due to channel error in a distributed decision fusion system is studied. It is shown that when delays

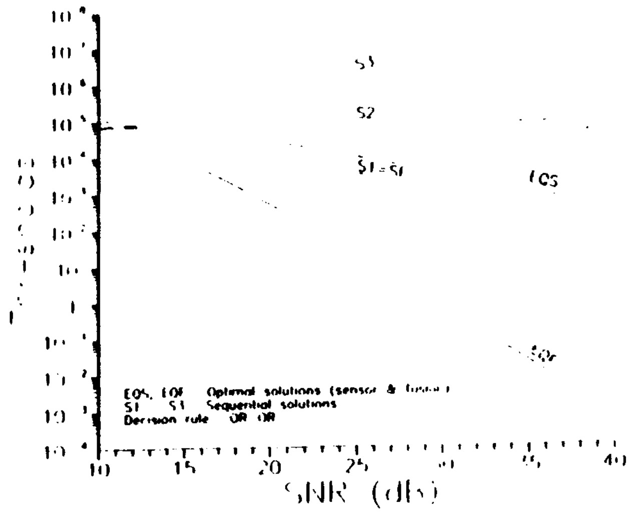


Fig. 6. Comparison of thresholds for $N = 3$, $p = 1$, and equal SNR case.

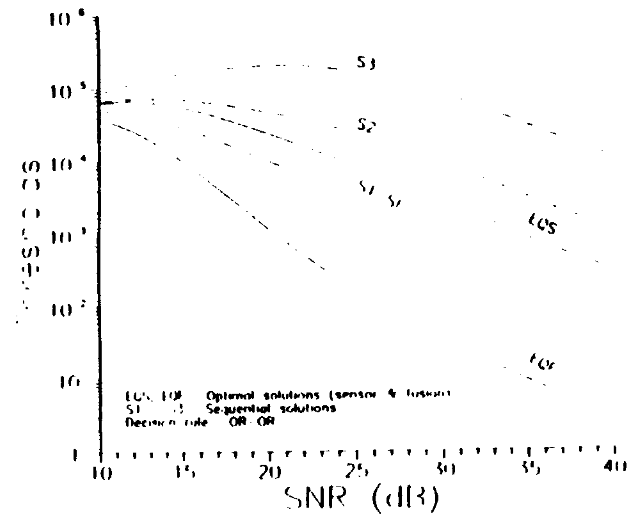


Fig. 7. Comparison of thresholds for $N = 3$, $p = 0.9$, and equal SNR case.

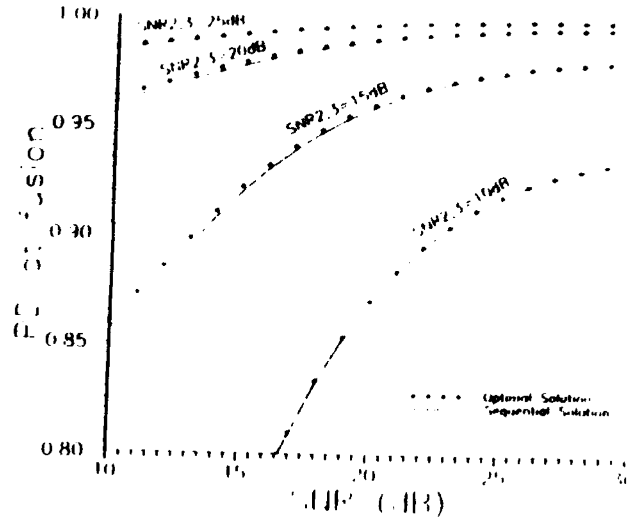


Fig 8. PD at fusion vs. SNR of the first sensor for $N = 3$ and $p = 0.9$. SNR2,3 represents the SNR of the 2nd and the 3rd sensor. Decision rule: OR-OR.

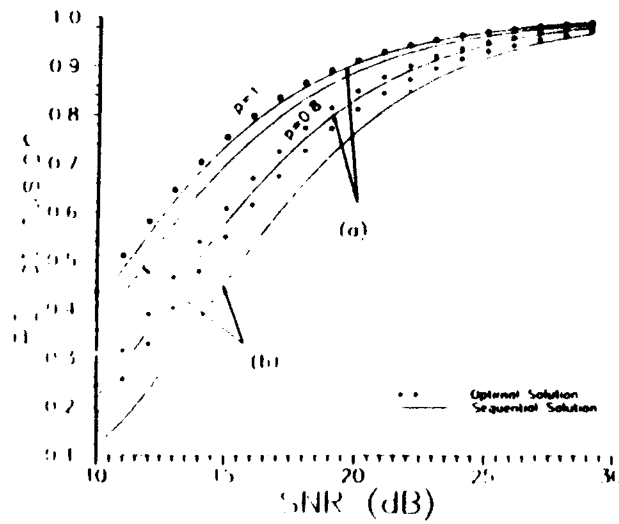


Fig. 9. PD at fusion for the case that $N = 3$, equal SNR, and channels have errors. Decision rule: AND-AND, $PF = 0.0001$. (a) $Pc = 0.0001$. (b) $Pc = 0.001$.

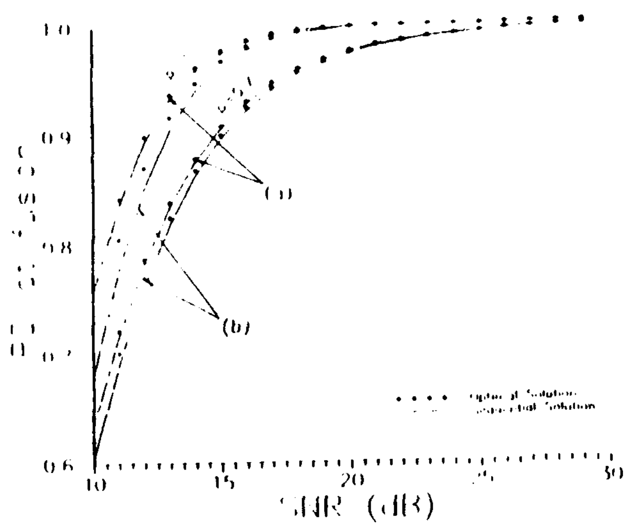


Fig. 10. PD at fusion for the case that $N = 3$, equal SNR and channels have errors. Decision rule: OR-OR, $PF = 0.0001$. (a) $Pc = 0.00001$. (b) $Pc = 0.00003$.

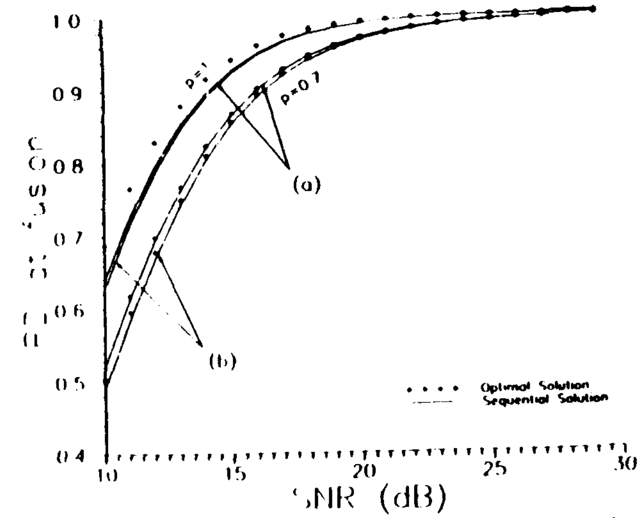


Fig. 11. PD at fusion for the case that $N = 3$, equal SNR and channels have errors. Decision rule: ML-OR, $PF = 0.0001$. (a) $Pc = 0.00001$. (b) $Pc = 0.00005$.

due to networking and errors due to channel noise are introduced in the distributed detection problem of Figure 1, the optimal fusion rule is still the Neyman-Pearson test at the fusion and Likelihood Ratio tests at all the peripheral sensors. However, the thresholds at the sensors and at the fusion center are adjusted to account for the networking delays and the channel errors. Using the Lagrange multipliers formulation, the optimal set of thresholds is obtained in terms of a set of coupled nonlinear equations whose solution depends on the unknown fusion rule and which thus cannot be solved analytically or numerically except in trivial cases. The suboptimal algorithm that was first derived to solve the distributed decision fusion problem assuming no delays and channel errors [11] was modified to solve for the thresholds sequentially in the presence of delays and channel errors. Numerical comparison of the optimal and suboptimal solutions for both equal and unequal SNR cases shows that the results given by the suboptimal algorithm are very close to the optimal ones.

It is shown that in the case of noisy channels the decision made by each sensor depends on the reliability of the corresponding transmission channel. Moreover, the probability of false alarm at the fusion is restricted by the channel errors. For a given decision rule, the error probability of any channel being in error must be kept at a certain level to achieve a desired probability of false alarm at the fusion. In the case of network delay, with or without channel errors, it is seen that the AND fusion rule yields poor performance. Furthermore, it suffers from a "folding effect," namely the performance of the fusion is not monotonic w.r.t. the values of the networking parameters (i.e., higher delay may yield higher probability of detection for the same probability of false alarm).

REFERENCES

1. R. R. Tenney and N. R. Sandell, Jr., Detection with distributed sensors, *IEEE Trans. Aerospace Electronic Syst.* AES-17:501–510 (July 1981).
2. E. Conte, E. D'Addio, A. Farina, and M. Longo, Multistatic radar detection: synthesis and comparison of optimum and suboptimum receivers, *IEEE Proc. F, Communication, Radar, and Signal Process*, 1983.
3. L. K. Ekchian, Optimal design of distributed detection networks, Ph.D thesis, Massachusetts Institute of Technology, 1983.
4. J. Tsitsiklis and M. Athans, On the complexity of distributed decision problems, *IEEE Trans. Automatic Control* AC-30(5):440–446 (May 1985).
5. F. A. Sadjadi, Hypothesis testing in a distributed environment, *IEEE Trans. Aerospace Electronic Syst.* AES-22:134–137 (March 1986).
6. Z. Chair and P. K. Varshney, Optimal data fusion in multiple sensor detection systems, *IEEE Trans. Aerospace Electronic Syst.*, AES-22(1):98–101 (Jan. 1986).
7. R. Srinivasan, Distributed radar detection theory, *IEEE Proceedings* 133 (Pt. F, 1):55–60 (Feb. 1986).

8. S. C. A. Thomopoulos, R. Viswanathan, and D. P. Bougoulias, Optimal decision fusion in multiple sensor systems, *IEEE Trans. Aerospace Electronic Syst.* 23(5):(Sept. 1987).
9. A. R. Reibman and L. W. Nolte, Design and performance comparison of distributed detection networks, *IEEE Trans. Aerospace Electronic Syst.* AES-23(6):789–797 (Nov. 1987).
10. R. Viswanathan, S. C. A. Thomopoulos, and R. Tumuluri, Optimal serial distributed decision fusion, *IEEE Trans. Aerospace Electronic Syst.* AES-24(4):366–376 (July 1988).
11. S. C. A. Thomopoulos, R. Viswanathan, D. K. Bougoulias, and L. Zhang, Optimal and suboptimal distributed decision fusion, in *Proceedings of the 22nd Annual CISS*, Princeton University, Princeton, New Jersey, March 16–18, 1988.
12. S. C. A. Thomopoulos and R. L. Zhang, Networking in distributed decision fusion, in *SPIE Proceedings on Sensor Fusion*, Vol. 876, January 1987, pp. 48–53.
13. S. C. A. Thomopoulos and R. L. Zhang, Distributed decision fusion with networking delays and channel errors, in *SPIE Proceedings on Sensor Fusion* 931:154–160, April 1988.
14. S. C. A. Thomopoulos, R. Viswanathan, and D. K. Bougoulias, Optimal distributed decision fusion, *IEEE Trans. Aerospace Electronic Syst.* AES-25(5):(Sept. 1989).

Received 11 October 1990; revised 5 June 1991

THEORIES IN DISTRIBUTED DECISION FUSION: COMPARISON AND GENERALIZATION

Stelios C. A. Thomopoulos[1]

Decision and Control Systems Laboratory
Department of Electrical and Computer Engineering
The Pennsylvania State University
University Park, PA 16802
Tel. (814) 865 - 3744; e-mail: sct@psuecl.bitnet

Abstract

Distributed Decision (Evidence) Fusion (DD(E)F) exhibits some interesting characteristics which are not present in centralized, or raw data, fusion. The interesting characteristics relate to the semantic information that the decisions (in the broader sense of the term) convey which (semantic information) is not present, at least explicitly, when raw data is fused. Different theories and results related to DD(E)F have appeared in the literature. Each theory takes a different stand on the definition of how to measure evidence or combine decisions. The objective of this paper is to investigate the nature of DD(E)F and establish a comparative basis between the two most prominent theories in DD(E)F, namely the Bayesian and Dempster-Shafer theories. To that extent, the similarities and differences between the two theories that result from the semantic differences in the format of the fused information are investigated. A performance comparison between the two theories is attempted. A Generalized Evidence Processing (GEP) theory that extends the Bayesian approach into fuzzy decision making is used to compare the performance of a Bayesian soft decision making system with that of a hard decision making Bayesian system. The similarities and differences between the GEP combining rule and the Dempster's combining rule are discussed and a consistency comparison between the two rules is performed.

1. Distributed Decision Fusion and Evidence Processing

Distributed Decision (Evidence) Fusion (or DD(E)F in the sequel) exhibits some interesting characteristics which are not present in centralized, or raw data, fusion. The interesting characteristics relate to the semantic information that the decisions (in the broader sense of the term) convey which is not present, at least explicitly, when raw data is fused. Different theories and results related to Distributed Decision Fusion (DDF) have appeared in the literature the last decade [TeSa 81, Sadj '86, ChVa '86, Srin '86, TVB '87, VTT '88, TVB '88, Demp '68, Shaf '76, Thom '90]. Each theory takes a different stand on the definition on how to measure evidence or combine decisions. The objective of this paper is to investigate the nature of DD(E)F, present some of the dominating theories on DDF and DEF, highlight similarities and differences among them that result from the semantic format of the fused information, and exploit natural topological equivalences between DDF and structures that exhibit learning abilities, such as neural networks.

To avoid concealing some of the issues under structural complexities and keep the discussion focused and as clear as possible we consider the simplest, yet fundamental, DDF topology and problem. We assume a parallel topology in which each sensor receives data from a common volume, Fig. 1. Furthermore, we assume that the sensors are perfectly aligned, so the problem of mismatch does not arise [ThOk '88]. In this parallel topology we assume the simplest DDF problem with each sensor's data statistically independent from the other sensors. Each sensor performs a local operation on its data and transmits the outcome to the fusion. The fusion collects all the local information from the sensors and produces the global inference. Several optimality results on Bayesian DDF have been obtained the recent years [TVB '89], [ChVa '86], [TVB '87], [VAT '89], [Thom '90], [Tsi '90]. Under the assuptions stated above, the optimal Bayesian DDF is shown in Fig. 2. In this paper we consider multi-level logic decision rules, in which the number of permissible local decisions exceeds the number of tested hypotheses. Decision rules for binary, as well as multiple hypotheses, testing problems are considered.

In DDF, the outcome of the global processing (fusion) depends on the outcome of the local data processing (sensor level) and the semantic format of the fused information. In the Bayesian context, the outcome of the local processing can be either hard decisions in a single-level logic [Thom '90], or soft decisions in a multi-level logic [Thom '90], or it can be the outcome of a simple quantization of the data, if no semantic attributes are attached to the outcome of the local processing [LLG '90]. In the context of the Dempster-Shafer's (D-S) theory, the outcome of the local processing is a set of probabilities that relate to the degree of support for each proposition in the frame of discernment by the the data of each local processor [Demp '68, Shaf '76]. Thus, the local processing outcome of a Bayesian DDF is a quantized scalar number, whereas the outcome of the D-S local processor is a real-valued vector that corresponds to an entire probability distribution.

In addition to semantic differences in the output of the local processors, there are also substantial differences in the communication requirements for transmitting the local information to the fusion. Even in the presence

1. This research is sponsored in part by SDIO/IST and managed by the Office of Naval Research under Contract N00014-86-k-0515.

of multi-level logic, the communication requirements for transmitting one out of, say, M integers is substantially lower than transmitting an M-dimensional real-valued vector. Hence, the communication requirements for the Bayesian DDF are substantially lower than the requirements of D-S DEF for the same number of data. Thus, a meaningful comparison between Bayesian and D-S DDF should either fix the available communication bandwidth to be the same for both approaches, or fix the fusion objectives to be common and study the communication overhead. In this paper we attempt a comparison of the D-S DDF with the Bayesian DDF assuming identical communication requirements.

Several optimality results on Bayesian DDF have been obtained the recent years [TVB '89], [ChVa '86], [TVB '87], [VAT '89], [Thom '90], [Tsi '90]. In [Thom '88 and Thom '90] a Generalized Evidence Processing (GEP) theory was introduced. The theory generalizes the Bayesian DDF into a framework where soft decision making is allowed. The GEP theory is briefly summarized in the next section. For a complete description of the GEP theory, see [Thom '90 and Thom '90].

2. Generalized Evidence Processing Theory

The pivoting idea behind GEP theory is the separation of hypotheses from decisions. Once this separation is understood, the Bayesian (or N-P) DDF theory can be extended to a frame of discernment similar to that of D-S theory. In the context of GEP theory, the choice of different decisions can be thought off as different quantization levels of the data. For notational simplicity, the GEP theory is first presented for binary hypothesis decision fusion. Generalization to multiple hypotheses decision fusion follows at the end of the section. Let H_1, H_0 be the two hypotheses under test. The probability space is partitioned into two regions according to the events $\{\omega = H_1\}$ and $\{\omega = H_0\}$ with associated probabilities $P_1 \geq 0$ and $P_0 \geq 0$ respectively, where $P_1 + P_0 = 1$. Let d_0, d_1, and $d_2 := d_{0v1}$ be a frame of discernment used by a decision maker to partition the probability space according to the gathered evidence, where the three decisions correspond to the propositions "H_0 true," "H_1 true," and "H_0 or H_1 true," respectively. The decision $d_2 := d_{0v1}$, where "v" stands for "or," indicates the inability of the decision maker to come up with conclusive evidence on the true nature of the hypothesis.

In the classical probabilistic (Bayesian) framework, the probability associated with d_{0v1} is equal to

$$\Pr[d_{0v1}] = \Pr[H_0 + H_1] = \Pr[H_0] + \Pr[H_1] = 1 \qquad (2.1)$$

since H_0 and H_1 constitute a disjoint coverage of the probability space over which the evidence processing problem is defined. As it was mentioned earlier, the apparent weakness of the Bayesian theory to incorporate non-mutually exclusive, i.e. redundant, propositions gave rise to the D-S theory which is particularly efficient in dealing with fuzzy propositions. However, by disassociating decisions from hypotheses, a unified framework is created which can accommodate both Bayesian and D-S DDFs.

In the context of GEP theory, the basic probability assignment (bpa) is accomplished either by minimizing a generalized Bayesian risk [Thom '89], or through any method that is applicable to D-S theory [Thom '90]. If the objective at the fusion is to minimize a generalized Bayesian risk, evidence combining in the GEP theory is done using likelihood ratio functions and pairwise multiplication of probabilities according to the way described in Table I and Eq. (2.2). The GEP combining rule involves pairwise multiplication of probability masses according to Table I as in D-S theory. However, in GEP theory, the masses are associated via thresholds in an optimal way so that a certain risk is minimized, or so that the probability of detection is maximized for fixed false alarm and indecision probabilities (generalized Neyman-Pearson test), whereas in D-S theory the probability masses (beliefs) are combined according to intersection of events, resulting in evidence conflict (Eq. 3.6). For a numerical study of the effect of the decision cost on the selection of the bpa and the performance of the GEP DDF rule see [Galu '90].

Table I ***GEP Evidence Combining Rule (2 hypotheses, 3 decisions)***

S2 / S1	$m_2^i(d_0)$	$m_2^i(d_1)$	$m_2^i(d_2)$
$m_1^i(d_0)$	$m_1^i(d_0)\, m_2^i(d_0)$	$m_1^i(d_0)\, m_2^i(d_1)$	$m_1^i(d_0)\, m_2^i(d_2)$
$m_1^i(d_1)$	$m_1^i(d_1)\, m_2^i(d_0)$	$m_1^i(d_1)\, m_2^i(d_1)$	$m_1^i(d_1)\, m_2^i(d_2)$
$m_1^i(d_2)$	$m_1^i(d_2)\, m_2^i(d_0)$	$m_1^i(d_2)\, m_2^i(d_1)$	$m_1^i(d_2)\, m_2^i(d_2)$

The probabilities in Table I are conditioned on each hypothesis i, i = 0, 1. Thus, each m_j^i, j = 1, 2, in Table I is a conditional probability for i = 0, 1. Hence, the initial probability combining takes place among conditional probabilities only. For $i = 0, 1$, each product term in Table I. is a probability mass on the LRT coordinate axis with abscissa $m_j^1(d) / m_j^0(d)$ for every $d = d_0, d_1, d_2$. Evidence combining under each hypothesis is done from Table I by summing the probabilities from Table I whose abscissae fall in specific intervals specified either by an optimization criterion, or a certain desired performance. Hence, for $d = d_0, d_1, d_2, \ldots, d_N$, evidence combining under each hypothesis H_i, $i = 0, 1$, is done according to the **threshold rule**

$$m_1^1(d_k)\, m_2^1(d_m) \rightarrow \text{decision } d_j \quad \text{if} \quad \frac{m_1^1(d_k)\, m_2^1(d_m)}{m_1^0(d_k)\, m_2^0(d_m)} \in F_j \tag{2.2}$$

where F_j is the decision region that favors decision d_j. The regions F_j may be determined so that a performance criterion is optimized at the fusion (and possibly at the sensors). For a single binary hypothesis, the decision regions at the fusion are determined by simple thresholds, in which case the decision rule (2.23) simplifies to

$$m_1^1(d_k)\, m_2^1(d_m) \rightarrow \text{decision } d_j \quad \text{if } t_j < \frac{m_1^1(d_k)\, m_2^1(d_m)}{m_1^0(d_k)\, m_2^0(d_m)} < t_{j+1} \tag{2.3}$$

for all k, m, and j, where t_j are the thresholds of the LRT's associated with the different decisions that minimize some risk function. If **multiple hypotheses** (more than two) are tested, the combining rule is extended to combine the belief functions of the individual sources at the fusion and generate the new conditional belief function under each hypothesis. The association of the new belief function at the fusion with the set of admissible decisions must be done by using the multiple-hypotheses LRT [VTre '68], or another test that optimizes some performance measure. It must again be underlined that the probabilities in the GEP combining rule need not be defined through Bayesian reasoning, but may very well correspond to belief functions resulting from the D-S approach.

In the multiple hypotheses case, the conditional belief function in GEP becomes a multi-variable function of the LRs $\left(\Lambda_k(d) := \prod_{j=1}^{J} \frac{dP(d_j \mid H_k)}{dP(d_j \mid H_0)},\ k = 1, 2, \ldots, m-1\right)$ where J is the number of sensors in the fusion system, d_j the decision of the j-th sensor, and m the number of tested hypotheses. The evidence from the different sensors is combined by forming the joint probability distribution of the LR's under each hypothesis, i.e. by generating $dP(\Lambda_1, \Lambda_2, \ldots, \Lambda_{m-1} \mid H_k)$, $k = 1, 2, \ldots, J$. For two sensors with independent decisions conditioned on each hypothesis, the conditional evidence combining rule of GEP for three hypothesis and soft decisions (fuzzy logic), can be implemented using Table II.

Table II Evidence combining rule for multiple hypotheses in GEP theory

(d_1, d_2)	$\Lambda_1(d_1, d_2)$	$\Lambda_2(d_1, d_2)$	$dP(\Lambda_1(d_1, d_2), \Lambda_2(d_1, d_2) \mid H_k)$ $= dP(\Lambda_1(d_1, d_2) \mid H_k)\, dP(\Lambda_2(d_1, d_2) \mid H_k)$ $= \prod_{j=1}^{2} dP(\Lambda_1(d_j) \mid H_k)\, dP(\Lambda_2(d_j) \mid H_k)$
(0, 0)	$\Lambda_1(0,0)$	$\Lambda_2(0,0)$	$dP(\Lambda_1(0,0) \mid H_k)\, dP(\Lambda_2(0,0) \mid H_k)$
(0, 1)	$\Lambda_1(0,1)$	$\Lambda_2(0,1)$	$dP(\Lambda_1(0,1) \mid H_k)\, dP(\Lambda_2(0,1) \mid H_k)$
(0, 2)	$\Lambda_1(0,2)$	$\Lambda_2(0,2)$	$dP(\Lambda_1(0,2) \mid H_k)\, dP(\Lambda_2(0,2) \mid H_k)$
(0, 0v1)	$\Lambda_1(0,0v1)$	$\Lambda_2(0,0v1)$	$dP(\Lambda_1(0,0v1) \mid H_k)\, dP(\Lambda_2(0,0v1) \mid H_k)$
(0, 0v2)	$\Lambda_1(0,0v2)$	$\Lambda_2(0,0v2)$	$dP(\Lambda_1(0,0v2) \mid H_k)\, dP(\Lambda_2(0,0v2) \mid H_k)$
(0v1, 0)	$\Lambda_1(0v1,0)$	$\Lambda_2(0v1,0)$	$dP(\Lambda_1(0v1,0) \mid H_k)\, dP(\Lambda_2(0v1,0) \mid H_k)$
(0v1, 1)	$\Lambda_1(0v1,1)$	$\Lambda_2(0v1,1)$	$dP(\Lambda_1(0v1,1) \mid H_k)\, dP(\Lambda_2(0v1,1) \mid H_k)$
(0v1, 2)	$\Lambda_1(0v1,2)$	$\Lambda_2(0v1,2)$	$dP(\Lambda_1(0v1,2) \mid H_k)\, dP(\Lambda_2(0v1,2) \mid H_k)$
(0v1, 0v1)	$\Lambda_1(0v1,0v1)$	$\Lambda_2(0v1,0v1)$	$dP(\Lambda_1(0v1,0v1) \mid H_k)\, dP(\Lambda_2(0v1,0v1) \mid H_k)$
(0v1, 0v2)	$\Lambda_1(0v1,0v2)$	$\Lambda_2(0v1,0v2)$	$dP(\Lambda_1(0v1,0v2) \mid H_k)\, dP(\Lambda_2(0v1,0v2) \mid H_k)$
(0v2, 1)	$\Lambda_1(0v2,1)$	$\Lambda_2(0v2,1)$	$dP(\Lambda_1(0v2,1) \mid H_k)\, dP(\Lambda_2(0v2,1) \mid H_k)$
..., etc			

Once all the entries in Table II are entered, the evidence is combined by adding the probabilities from the fourth column together when the corresponding abscissae, i.e. the pairs $(\Lambda, (d, d))$, $\Lambda, (d, d))$ in the second and third columns, are identical. Once the evidence from all sensors is combined using tables similar to Table II, decisions are associated with the combined evidence using rule (2.23) so that a desired performance criterion is optimized.

Thus, evidence combining at the fusion is done conditioned on each hypothesis separately. The evidence is then associated with the admissible decisions unconditionally using a LRT or a test that optimizes some performance measure. Notice that the set of decisions need not be the same as the set of hypotheses. Thus, **evidence combining and decision making are understood as separate concepts in the framework of the Generalized Evidence Combining Theory**.

The generalization of the Bayesian (and N-P) theory by the GEP theory is straightforward. An interpretation is probably required to establish the correspondence between GEP and D-S theories. If the probabilities $P(u_k = i \mid H_j)$, $i = 1, 2, 3$, are considered as (conditional) bpa's (basic probability assignments [Shaf '6]) in the D-S theory for the k-th sensor, $k = 1, 2, \ldots, N$, under hypothesis H_j, $j = 0, 1$, the evidence from the different sensors at the fusion is combined using the conditional distribution of the LR under the different hypothesis according to Table I or II. A new (conditional) belief function is generated using the decision thresholds at the fusion. The (hard) decisions at the sensors are used to simply produce a hard decision at the fusion, if needed, according to some optimality criteria. In *that respect, the GEP theory not only* defines and processes the evidence according to an a-priori set of optimality criteria, but also provides, if needed, for optimized hard decisions both at the local (sensor) as well as global (fusion) level, a capability which is not built-in the D-S theory (see Section 3).

The decision boundaries in GEP theory determine how evidence is associated with propositions at the fusion and reflect the choice of the costs w_{ij}. To demonstrate the effect that the semantic content of the local decisions has on the global decision (fusion), several experiments were conducted in Gaussian and slow-fading Rayleigh channels. The *following statistical model were assumed for the two channels.*

Gaussian: *Observation model at each sensor:* $r \sim G(0,1) : H_0$, and $r \sim G(s,1) : H_1$, where $G(\alpha,\beta)$ designates an α *mean and variance* β Gaussian distribution. If P_F is the operating false alarm probability, the associated threshold $t_b := Q^{-1}(P_f)$, where $Q(\,) = 1 - \Phi(\,)$ is the cumulative distribution function (cdf) of the standard normal, and Q^{-1} is its inverse.

Rayleigh: *False alarm probability:* $P_F = [\lambda(1+\epsilon)]^{-(1+\frac{1}{\epsilon})}$; ***Detection probability:*** $P_D = [P_F]^{(\frac{1}{1+\epsilon})}$

where λ is the threshold used, and ϵ the SNR at the sensor. In the single-level local logic Bayesian DDF with hard decisions at the sensors and fusion, the probabilities at the sensors were generated assuming fixed false alarm probabilities at the sensors equal to 0.05. For the multi-level local logic DDF, the ambiguous (soft or "fuzzy") decisions were generated by considering a ±20% uncertainty region about the thresholds that determine the decision boundaries in the Bayesian case. The numerical results that are presented refer to the binary hypothesis testing *from* which the set of "soft" decisions consists of $\{d_0 = H_0, d_1 = H_1, d_2 = H_0 \vee H_1\}$. Additional results for ternary hypothesis testing and arbitrary probability assignments can be found in [Galu '90].

In a set of experiments, the performance of Bayesian DDF (i.e. GEP) with soft decisions at the local level and hard decisions at the fusion was compared to Bayesian DDF with hard decisions both locally and at the fusion. Using the "±20% uncertainty region" described above to generate the soft decision "H_1 or H_0," the *Level Of Confidence* (*LOC*), which *is equivalent to the (unconditional) probability of* correct decision, was used for comparison. The LOC curves in Fig. 3 indicate that GEP outperforms Bayesian DDF with hard local decisions in all cases. The curves were obtained by assuming a fixed false alarm probability 0.05 at the sensors and 0.005 at the fusion.05. GEP outperforms hard-decision Bayesian DFF in both binary and ternary hypothesis testing, in both Gaussian and slow-fading Rayleigh channels and for any number of sensors. This does not come as a surprise if the decision set of GEP is thought of as the result of multi-level quantization of the data, and the quantization is done according to a semantically intuitive fashion.

3. Distributed Decision Fusion using Dempster-Shafer's Theory

The difference between the Bayesian and D-S theory lies on the type of information that each sensor transmits to the fusion after processing the data locally. As it will become clear in the sequel, if the propositions in the D-S theory are identified with decisions in the GEP (Generalized Bayesian) theory, then there are no semantic differences in the frame of discernment between the two theories. The difference lies on that the probability assignment in GEP still satisfies the Bayesian rule, whereas the evidence assignment does not. Assuming that the number of hypotheses that are tested is fixed and the number of decisions (or frame of discernment in the D-S terminology) is fixed, the output of the *local data processing is a set of probabilities regarding the* likelihood that the data have been generated by one of the particular hypotheses or subset of hypotheses according to the frame of discernment. To that extend, the use of the term decisions in the D-S theory does not precisely reflect the output of the local processing. It is more appropriate to characterize the outcome of the local processing as evidence about a chosen set of proposition rather than decision regarding a specific hypothesis or set of hypotheses. Thus, even if the frame of discernment is kept common between

Bayesian and D-S approaches (by utilizing multi-level Bayesian logic), the mapping of the data in the output of the local processor is completely different: the Bayesian processor maps each data to a particular, single decision (integer-valued scalar), whereas the D-S processor maps the same data to a set of probabilities (multidimensional real-valued vector) associated with all decisions in the frame of discernment. Hence, the communication requirements between Bayesian and D-S processors and fusion are different. Assuming a frame of discernment consisting of k propositions, the communication requirements for the Bayesian case is 2logk (the bandwidth required to transmit one of k bits), whereas for the D-S processor k analog outputs must be transmitted to the fusion. Thus, unless the communication requirements for the two approaches are made common, no direct comparison in the performance of the two schemes is meaningful. Since such a performance is beyond the objectives of this paper, we limit the discussion in the structure of the D-S DDF.

In D-S theory, a set of mutually exclusive and exhaustive propositions $u_1, u_2, \ldots, u_m$ is assumed toward which evidence is being offered. To each proposition, their disjunctions, and negations, a nonnegative number between zero and one (or probability mass) is assigned. If A is an atomic proposition, a disjunction of propositions, or a negation of a proposition, then a probability mass, m(A), is assigned to A. The quantity m(A) is a measure of the belief in proposition A based on the evidence offered. If U designates the frame of discernment, then

$$\sum_{A \in U} m(A) \leq 1 \tag{3.1}$$

with the remaining $1 - \sum_{A \in U} m(A)$ mass attribute to ignorance. Assuming that ignorance constitutes a separate proposition and extending the set U to include this proposition, expression (3.1) holds as an equality. According to D-S theory, a support function is defined for single propositions as

$$spt(u_i) = m(u_i) \tag{3.2}$$

and for more complex propositions as

$$spt(A) = \sum_{B \subset A} m(B) \tag{3.3}$$

where "C" indicates subset. The plausibility function is defined as

$$pls(u_i) = 1 - spt(\bar{u}_i) \tag{3.4}$$

where $\bar{u}_i$ indicates the negation of proposition u_i. Alternatively, the plausibility function for a proposition u_i is obtained by summing the masses of all the disjunctions that contain u_i, including itself, i.e.

$$pls(u_i) = \sum_{u_i \in A} m(A) \tag{3.5}$$

Hence, the support function is indicative of how much evidence is offered in support of a given proposition by all the propositions that relate to it. Furthermore, the plausibility function is indicative of how likely it is for a given proposition to have generated the data.

Evidence from different, and *independent*, sources defined over the same frame of discernment, is fused according to Dempster's combining rule [Depm '68]

$$m(u_i) = m_1 \oplus m_2 = \frac{\sum_{A_i B_j = u_i} m_1(A_i)\, m_2(B_j)}{1 - \sum_{A_k B_m = \phi} m_1(A_k)\, m_2(B_k)} \tag{3.6}$$

where m_1 and m_2 designate the support (belief) functions from the two different sources of evidence defined over the same frame of discernment, u_i is the proposition toward which evidence is sought, and "ϕ" is the empty set [Shaf '76]. Renormalization of the combined evidence in rule (3.6) is required to reject evidence that corresponds to conflicting propositions. The D-S combining rule can be implemented in a tabular fashion that resembles that of GEP theory [Thom '89, '90]. To illustrate the mechanical similarities that exist between the Dempster's combining rule and the GEP DDF, consider a simple binary hypothesis testing problem. If the frame of discernment is defined as $\{u_0 = H_0, u_1 = H_1, u_2 = H_0$ or $H_1\}$, with u_2 indicating the inability to associate evidence from the data with a definite hypothesis, the Dempster's combining rule for two sensors can be implemented using Table III. In Table III, k designates evidence associated with conflicting propositions which is used as normalizing factor in (3.6). The combined evidence is calculated by summing all the product terms from Table III that result to the same intersection proposition, and normalizing the result. In multiple-source evidence combining, rule (3.6) is repeated sequencially until the evidence from all sources is exhausted.

Table III Dempster's Combining Rule

S2 / S1	$m_2^1(u)$	$m_2^1(u)$	$m_2^1(u)$
$m_1(u)$	$m(u)=m_1(u)\,m_2(u)$	$k=m_1(u)\,m_2(u)$	$m(u)=m_1(u)\,m_2(u)$
$m_1(u)$	$k=m_1(u)\,m_2(u)$	$m(u)=m_1(u)m_2(u)$	$m(u)=m_1(u)\,m_2(u)$
$m_1(u)$	$m(u)=m_1(u)m_2(u)$	$m(u)=m_1(u)m_2(u)$	$m(u)=m_1(u)\,m_2(u)$

The difference between the D-S and Bayesian theory is that the probability assignments for the propositions in the frame of discernment of the D-S theory do not satisfy the fundamental axiom of (Bayesian) probability, namely

$$P(A+B) = P(A) + P(B) - P(AB) \tag{3.7}$$

In the D-S context, the proposition A+B is viewed as a separate entity in the frame of discernment and can be assigned an arbitrary probability mass. Still all the probability assignments in the D-S theory must add up to one or some positive quantity less than one, with the remaining probability mass to add dp to one attributed to total ignorance [Shaf '76]. A correspondence between the propositions as defined in the D-S theory and the decisions as defined in the multi-level logic Bayesian theory can be established if the decisions of the multi-level logic Bayesian framework are identified with the propositions in the D-S *frame of discernment*. Once this correspondence is established the fusion performance under the two approaches can be studied under common communication constraints. By disassociating decisions from the hypotheses under test, the Generalized Evidence Processing (GEP) provides a semantically common framework within which the N-P and D-S DDF approaches can be compared under common communication constraints.

Due to the difference in the way evidence is generated in Bayesian (N-P) and D-S theory, an unconditional performance comparison between the two theories is not, in general, feasible. Since in a lot of practical applications the performance of a decision making system is determined by fixing the false alarm probability and maximizing the detection probability at the fusion, it is meaningful to compare the Bayesian and D-S approach based on an N-P criterion. In order to make the comparison possible, we assume that the basic probability assignment of the D-S DDF at the local level is determined using the likelihood function, i.e. we assume that

$$m(\alpha \mid r) = P(\alpha \mid r) \tag{3.8}$$

where α designates a proposition towards which evidence is provided, and r the observations. Even when the bpa is resolved at the local level, the decision rule at the fusion after the local evidence is combined remains undetermined. In order to keep the decision rule in a D-S context while maintaining a basis for comparison with the Bayesian DDF, the decision rule that will be used for the D-S DDF will assign the data to the proposition that has the highest support among all propositions in the frame of discernment that correspond to definite hypotheses, i.e.

$$d(r) := d_i(r) : \max_i \operatorname{sdp}(d_i) \text{ and } d_i = H_i, \text{ i over all single hypothesis propositions} \tag{3.9}$$

With the above assumptions, we prove the following theorem.

Theorem 1 Assume that the objective of the fusion is to maximize the detection probability after fusion for fixed false alarm probability. Let the observations of the local sensors be independent from each other conditioned on each hypothesis. Let the bpa for the D-S DDF be determined by the likelihood function (3.8) at the local level. If the fusion rule is the rule (3.9) above, then:

(a) if the local frame of discernment coincides with the hypotheses under test, i.e. no unions of hypotheses are used as basic propositions, the performance of the D-S DDF is the same as the centralized N-P (Bayesian) fusion.

(b) if compound-hypotheses propositions are allowed in the local bpa, then the performance of the D-S DDF is always inferior to the centralized N-P fusion and the distributed N-P fusion for the same communication overhead.

Proof We prove the theorem for the case of two sensors and binary hypotheses testing. A generalization of the proof, although notationally involved, does not present any conceptual difficulties and as such is omitted.

Part (a) According to the assumptions of the theorem, the bpa is

$$m(H_i) := \Pr(H_i \mid r) = [p(r \mid H_i)\Pr(H_i)] / p(r) \; : i = 0, 1 \tag{3.10}$$

and so the D-S requirement

$$m(H_1) + m(H_0) = 1 \tag{3.11}$$

is satisfied. Using the Dempster's combining rule ( ) for two sensors, we obtain

$$\sup(H_1) = [m^1(H_1)m^2(H_1)] / [1 - m^1(H_1)m^2(H_0) - m^1(H_0)m^2(H_1)] \tag{3.12}$$

where the division is the result of renormalization due to the existence of conflicting evidence mass after fusion, and the superscripts identify the sensors. A similar expression is obtained for the H_0 hypothesis if the indexes in ( ) are switched. The proposed decision rule (3.9) translates to

$$sup(H_1) / sup(H_0) \underset{H_0}{\overset{H_1}{\gtrless}} t \tag{3.13}$$

were t is some threshold to be determined. Taking into account that for this particular case the D-S rule yields

$$sup(H_i) = m(H_i) \tag{3.14}$$

and using expression (3.3), the D-S decision rule gives after some elementary algebra

$$[p(r_1 | H_1)p(r_2 | H_1)] / [p(r_1 | H_0)p(r_2 | H_0)] \underset{H_0}{\overset{H_1}{\gtrless}} t \tag{3.15a}$$

or

$$[p(r_1 | H_1)p(r_2 | H_1)] / [p(r_1 | H_0)p(r_2 | H_0)] \underset{H_0}{\overset{H_1}{\gtrless}} t \tag{3.15b}$$

or

$$[p(r_1 | H_1)/p(r_1 | H_0)] [p(r_2 | H_1) / p(r_2 | H_0)] \underset{H_0}{\overset{H_1}{\gtrless}} t \tag{3.15c}$$

or

$$[p(r_1 | H_1)p(r_2 | H_1) - t\, p(r_1 | H_0)p(r_2 | H_0)] \underset{H_0}{\overset{H_1}{\gtrless}} 0 \tag{3.15d}$$

which is precisely the centralized Bayesian N-P test. Thus, the performance of the D-S DDF in this case is identical to the optimal centralized Bayesian DDF for the same false alarm probability at the fusion.

Part (b) In the binary hypotheses testing case the only compound proposition in the frame of discernment is {H_0 or H_1 }. If we assume, without loss of generality, that the bpa for the three propositions is done by subtracting an equal amount of probability from the two propositions that correspond to the definite hypotheses and associating it with the compound proposition, the following bpa results

$$m_i(H_0) = Pr(H_0 | r_i) - \epsilon(r_i)/2$$
$$m_i(H_1) = Pr(H_1 | r_i) - \epsilon(r_i)/2 \tag{3.16}$$
$$m_i(H_0 \text{ or } H_1) = \epsilon(r_i) := \epsilon_i$$

where the probability mass $\epsilon(r_i)$ can be data dependent. Using the Dempster's combining rule to fuse the evidence and suppressing the explicit dependence of ϵ_i on the data for notational simplicity, we obtain the following expressions for the support function regarding the two hypotheses.

$$sup(H_0) = [m_1 (H_0)m_2 (H_0) + 1/2[\epsilon_1 m_2 (H_0) + \epsilon_2 m_1 (H_0)] - 3\epsilon_1 \epsilon_2 /4] / [1 - \text{conflicting evidence}] \tag{3.17a}$$

and

$$sup(H_1) = [m_1 (H_1)m_2 (H_1) + 1/2[\epsilon_1 m_2 (H_1) + \epsilon_2 m_1 (H_1)] - 3\epsilon_1 \epsilon_2 /4] / [1 - \text{conflicting evidence}] \tag{3.17b}$$

from which the assumed decision rule

$$sup(H_1) / sup(H_0) \underset{H_0}{\overset{H_1}{\gtrless}} t \tag{3.18}$$

yields

$$[p(r_1 | H_1)p(r_2 | H_1) - tp(r_1 | H_0)p(r_2 | H_0)]$$
$$+ 1/2\{[\epsilon_2 p(r_1 | H_1)+\epsilon_1 p(r_2 | H_1)] - t[p(r_1 | H_0)+p(r_2 | H_0)]\} \underset{H_0}{\overset{H_1}{\gtrless}} 3\epsilon_1 \epsilon_2 /4[1-t] \tag{3.19}$$

By comparing the decision rule (3.19) with the optimal N-P test rule (3.15d), it is seen that the first term in brackets in the left side of (3.19) is identical to the term in the left side of (3.15d). Since the decision rule (3.15d) is ***the*** optimal decision rule in the N-P sense, rule (3.19) would achieve optimal performance *if and only if* the rest of the terms in (3.19) could be made identically equal to zero for a fixed threshold t. However, even with data dependent bpa assignment $\epsilon_1(r)$, this is not possible in general[2]. Thus, the performance of the D-S DDF is inferior to the optimal centralized N-P fusion. Furthermore, since the performance of the distributed N-P decision fusion can be arbitrarily close to the optimal centralized one [TVB '87, Thom '90] by simply including some additional quality information bits along with the decisions or by increasing the number of quantization levels, the performance of the N-P DDF is always superior to the performance of the D-S DDF for a lesser amount of communication requirements. Notice that in the D-S either the data itself has to be transmitted from the sensors to the fusion (which is the most efficient way), or the bpas must be transmitted thus making the communication requirements proportional to the number of propositions in the frame of discernment. (Clearly, a quantized version of the data or bpas can be transmitted resulting in reduction of communication requirements and performance as well.)

The above arguments extend easily to multiple sensor case. The general multi-hypothesis case can be handled in a similar way as the two hypothesis case, only the expressions become more complicated. □

To compare the consistency of GEP and D-S evidence combining rules (2.2, 2.3) and (3.6) respectively, the following experiment was conducted. Numerical results have been obtained for binary and ternary hypothesis testing, and for distribution based as well as arbitrary bpa's. However, due to limited space, results from the binary hypothesis testing will be presented only. For additional results, the reader is referred to [Galu '90 and Ga '90]. The binary hypothesis testing results will be presented first. For GEP, conditional probabilities at the fusion center were obtained in the same manner as in previously discussed simulations. The conditional probabilities at the sensor, from the GEP stimulation, were used as the original probability assignments at the sensor for the D-S theory simulation. Conditional probability masses were calculated at the fusion using Dempster's combining rule. The conditional probabilities from GEP and the conditional probability masses from D-S theory were then used to calculate conditional plausibility according to (3.5). The results were obtained for a false alarm probability of .05 at the sensor and .005 at fusion.

Figures 4 and 5 display results for Gaussian and Rayleigh distributed signals respectively. Both graphs show the plausibility conditioned on hypothesis H_1 for five and ten sensors. To compare the two combining rules for consistency, we define the crossover point as the SNR level above which the plausibility for the correct hypothesis, H_1, becomes greater than that for the incorrect hypothesis, H_0. Observe that for both the five and ten sensor cases the crossover point occurs at a lower SNR for GEP than for D-S theory. So GEP works correctly for a wider range of SNR than does D-S theory. Also notice the behavior as the number of sensors increases from five to ten. For GEP the crossover point moves to lower SNR while for D-S theory it does not move at all. This indicates that we can improve the performance of GEP by increasing the number of sensors, which is a very desirable feature. The performance of D-S theory, on the other hand does not improve when the number of sensors increases.

Figures 6, 7 show unconditional plausibility plots for the Gaussian and Rayleigh cases. More specifically they show the unconditional plausibility for the correct and incorrect hypotheses. Once again the results are shown for both five and ten sensors. We see that for all cases the plausibility for the correct hypothesis is higher at lower SNR for GEP than that for D-S theory. The separation between plausibility for correct and incorrect hypotheses is much clearer for GEP. In fact at very low SNR D-S theory fails to separate the plausibility for the correct hypothesis from that of the incorrect.

Conclusions

The two major evidence processing theories, namely Bayesian and Dempster-Shafer's, are presented as applied to the problem of Distributed Decision or Evidence Fusion. Some of the fundamental results in Bayesian and Neyman-Pearson DDF are presented. It is shown that a generalization of the Bayesian DDF using multi-level logic at the local processor can provide a framework that allows comparison of the performance of the Bayesian and D-S DDFs under certain conditions. To that extend, a theorem is developed that shows that if the objective is to maximize the detection probability at the fusion for fixed false alarm probability, the Bayesian DDD outperforms the D-S DDF when multi-level logic is used locally, i.e. at the sensors.

References

[Demp '68] Dempster, A. P., "A Generalization of Bayesian Inference," Journal of the Royal Statistical Society, Vol. 30, 1968, pp. 205-247.

[VTre '68] Van Trees, H. L., Detection, Estimations, and Modulation Theory, Vol. 1, John Wiley & Sons, New York, 1968.

[Shaf '78] Shafer, G. A., A Mathematical Theory of Evidence, Princeton University Press, 1976.

[TeSa '81] Tenney, R. R. and Sandell, N.R., Jr., "Detection with Distributed Sensors," IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-17, July 1981, pp. 501-510.

2. In the binary hypothesis testing, this is feasible if the mass that is associated with the compound decision $\{H_0$ or $H_1\}$ is removed entirely from the probability mass of one and only one of the two other definite decisions.

[TsAt '85] Tsitsiklis, J., and Athans, M., "On the Complexity of Distributed Decision Problem." IEEE Trans. on Automatic Control, AC-30, Vol. 5, May 1985, pp. 440-446.

[Sadj '86] Sadjadi, F. A., "Hypothesis Testing in a Distributed Environment." IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-22, March 1986, pp. 134-137.

[ChVa '86] Chair, Z. and Varshney, P.K., "Optimal Data Fusion in Multiple Sensor Detection Systems." IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-22, No. 1, January 1986, pp. 98-101.

[Srin '86] Srinivasan, R., "Distributed Radar Detection Theory." IEEE Proceedings, Vol. 133, Pt.F, No. 1, February 1986, pp. 55-60.

[TVB '87] Thomopoulos, S. C. A., Viswanathan, R. and Bougoulias, D.K., "Optimal Decision Fusion in Multiple Sensor Systems." IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-23, No. 5, Sept. 1987.

[TBZ '88] Thomopoulos, S. C. A., Bougoulias, D.K., and Zhang, L., "Optimal and Suboptimal Distributed Decision Fusion." SPIE Proceedings, Vol. 931, Sensor Fusion (1988), pp. 26-30.

[VTT '88] Viswanathan, R., Thomopoulos, S. C. A., and Tumuluri, R., "Optimal Serial Distributed Decision Fusion." IEEE Transactions on Aerospace and Electronic Systems, Vol. 24, No. 4, pp. 366-376, July 1988.

[VAT '88] Viswanathan, R., Ansari, A., and Thomopoulos, S. C. A., Optimal Partitioning of Observations in Distributed Detection." Abstracts of Intern. Symposium on Information Theory, Kobe, Japan, June 19-24, 1988, p. 195.

[ThOk '88] Thomopoulos, S. C. A., and Okello, N. N., "Distributed Detection with Mismatched Senors." SPIE '88 Symposium on Advances in Intelligent Robotic Systems, Boston, Nov. 1988.

[TVB '89] Thomopoulos, S. C. A., Viswanathan, R. and Bougoulias, D.K., "Optimal and Suboptimal Distributed Decision Fusion." IEEE Transactions on Aerospace and Electronic Systems, Vol. 25, No. 5, Sept. 1989.

[Thom '90] Thomopoulos, S. C. A., "Sensor Integration and Data Fusion," Special Issue on Sensor Integration and Data Fusion," International Journal of Robotics, Vol. 7, No. 3, 1990, pp. 337-372.

[Thom '90] Thomopoulos, S. C. A., "Theories in Distributed Decision Fusion," Ju;y 1990, submitted for publication.

[Galu '90] Galuza, M., Performance Evaluation of Generalized Evidence Theory in Distributed Detection Problems, M.S. Thesis, Dept. of Electr. & Comp. Engr., Penn State Univ., May 1990.

[Tsi '90] Tsitsiklis, J. N., "External Properties of Likelihood Ratio Quantizers." 1990 preprint.

[LLG '90] Longo, M., Lookabaugh, T. D., and Gray, R. m., "Quantization for Decentralized Hypothesis Testing under Communiation Constraints," IEEE Trans. on Info. Th., IT-36, No. 2, March 1990, pp. 241-255.

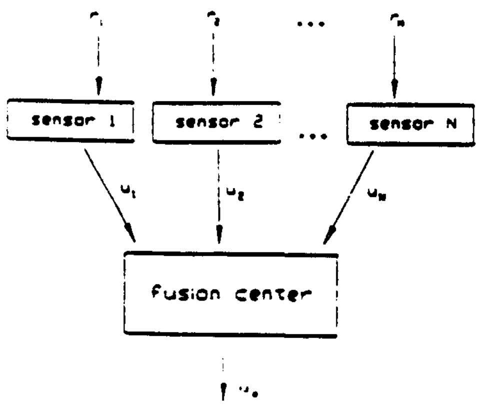


Figure 1 Parallel Sensor Topology

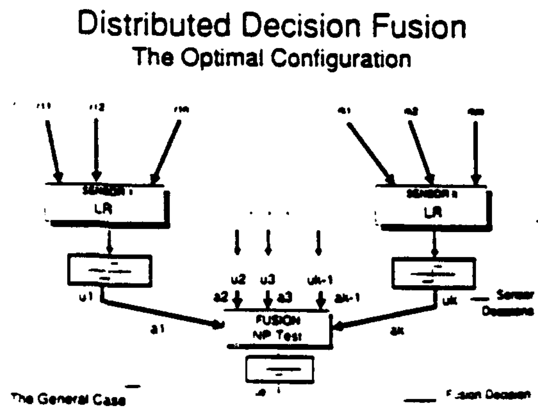


Figure 2 Optimal Bayesian DDF

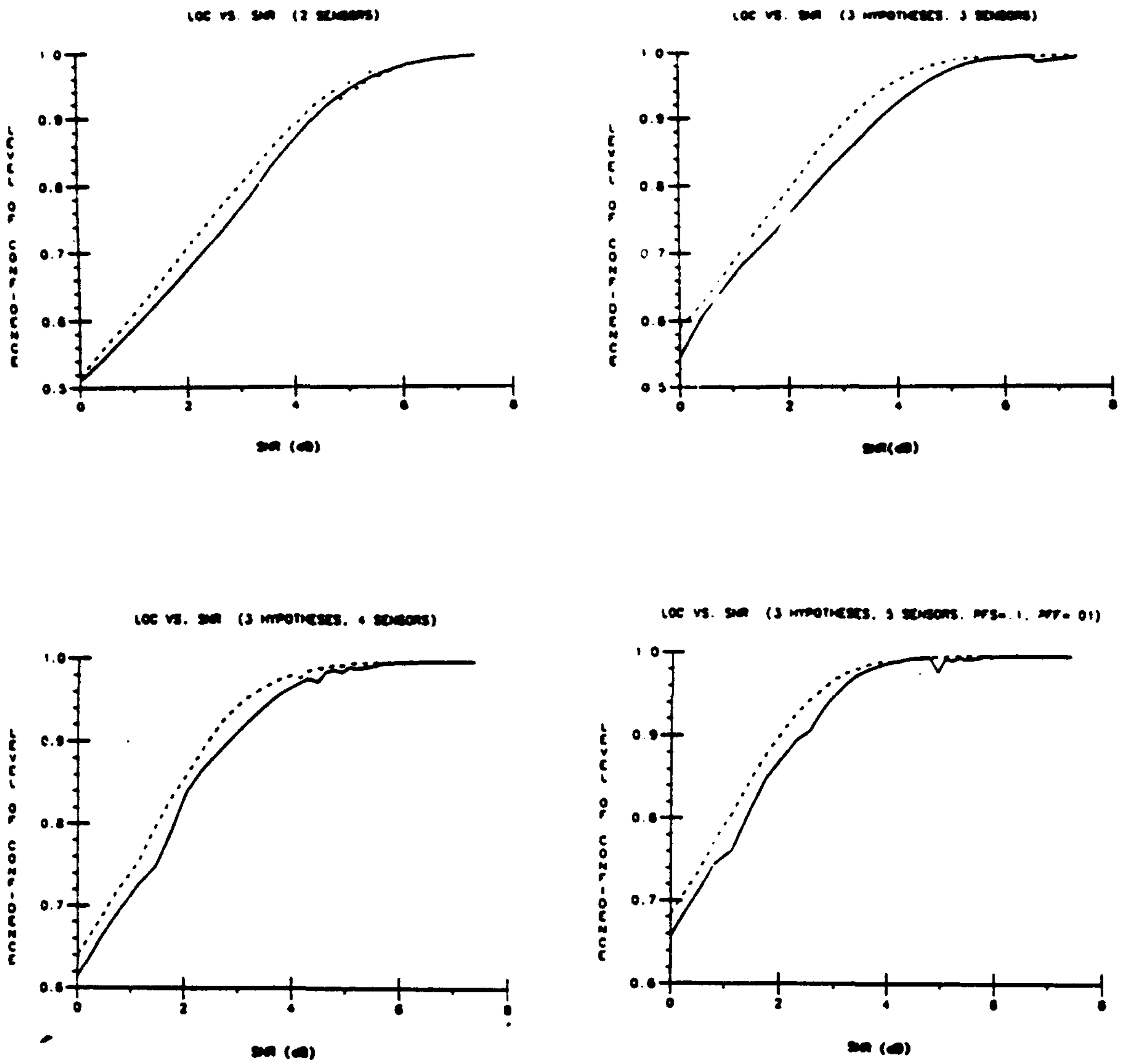


Figure 3 Level of Confidence (LOC) obtained using hard decisions (Bayesian) and soft decisions (GEP) in three hypotheses decison making. In both cases, the fusion makes hard decisions. LOC corresponds to total probability of deciding correctly. The solid (__) line curves correspond to Bayesian approach. The dashed (----) line curves correspond to GEP aproach. The different figures correspond to 2, 3, 4, and 5 sensors respectively. In all cases, GEP outperforms Bayesian.

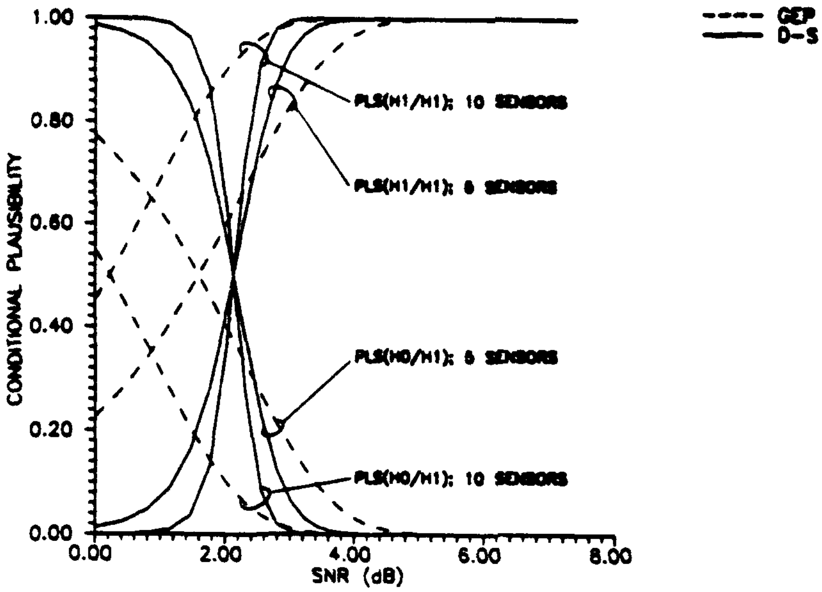


Figure 4 Conditional plausibility vs. SNR using GEP DDF and D-S DDF for 5 and 10 sensors; Gaussian case.

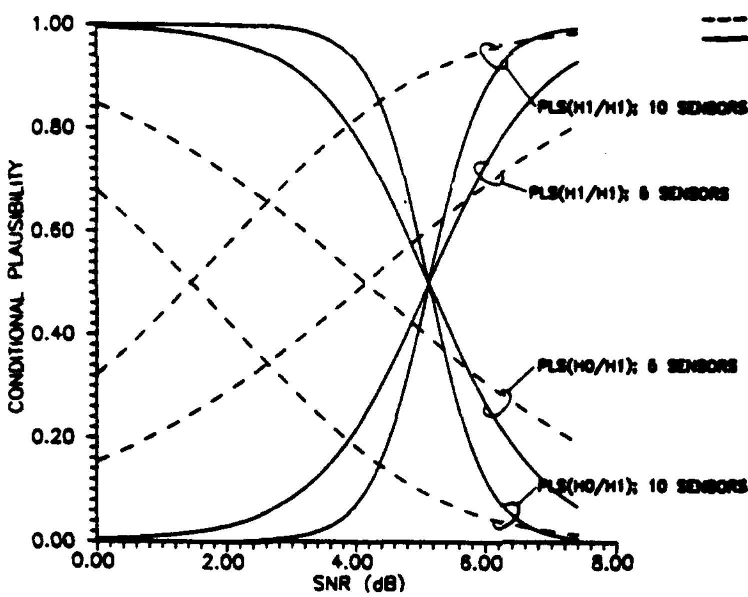


Figure 5 Conditional plausibility vs. SNR using GEP DDF and D-S DDF for 5 and 10 sensors; Rayleigh case.

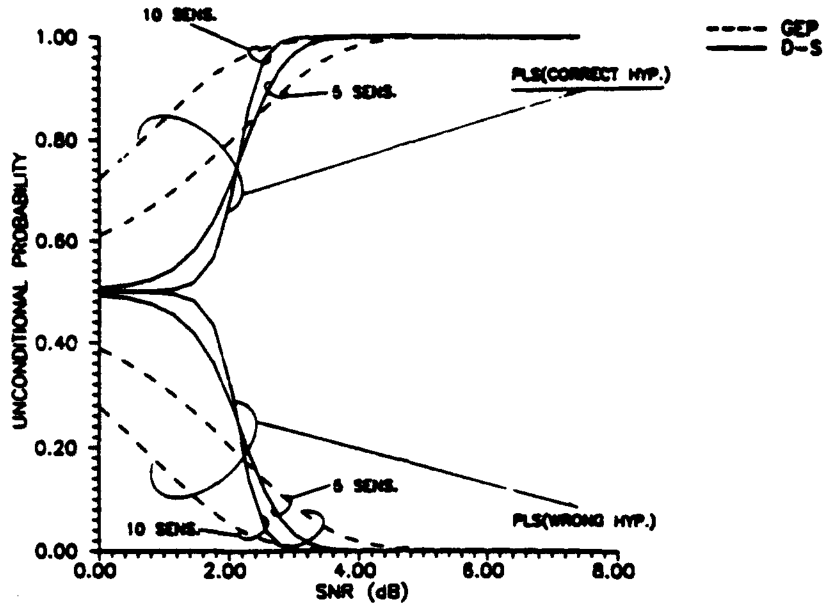


Figure 6 Unconditional plausibility vs. SNR using GEP DDF and D-S DDF for 5 and 10 sensors; Gaussian case.

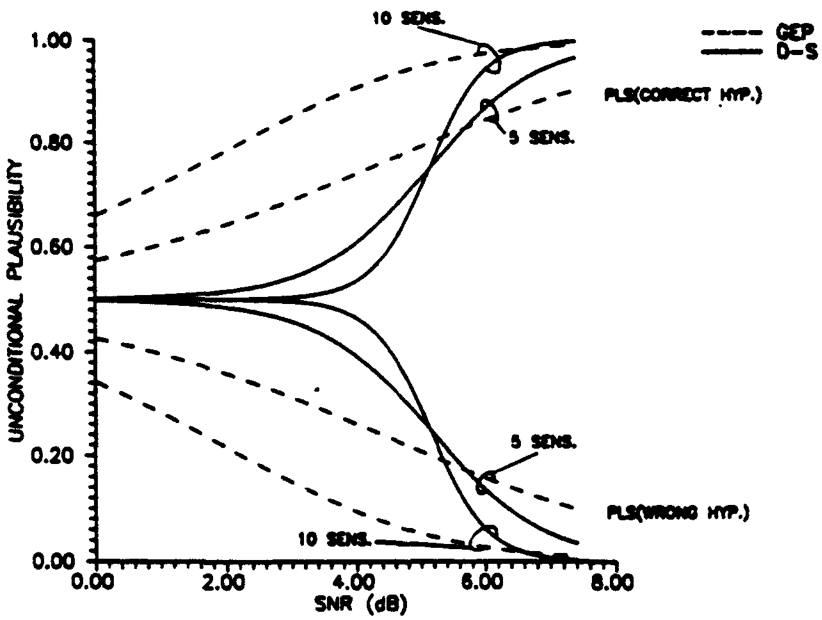


Figure 7 Unconditional plausibility vs. SNR using GEP DDF and D-S DDF for 5 and 10 sensors; Rayleigh case.

OBJECT TRACKING FROM IMAGE SEQUENCES USING STEREO CAMERA AND RANGE RADAR

Stelios C. A. Thomopoulos[1,2] **and Lars Nillson**

Information Processing and Intelligent Systems (IPIS) Laboratory
Department of Electrical Engineering
Southern Illinois University
Carbondale, IL 62901

Abstract - The problem of estimating the position of and tracking an object undergoing 3-D translational and rotational motion using passive and active sensors is considered. The passive sensor used in this study is a stereo camera, whereas the active is a range radar. Three different estimation approaches are considered. The first involves estimation of the object position by direct registration of stereo images. In the second approach, the Extended Kalman Filter is used for estimation with measurements the stereo images. In the third approach, an integral filter based on stereo images and range radar measurements is used for tracking. The three different approaches are compared via simulation in the tracking of an object undergoing a 3-D motion with random translational and angular accelaration.

1. INTRODUCTION

Object positionning and tracking using data from passive sensors, such as cameras, infrared (IR) sensors, etc, is a common problem in robotics, automated manifacturing, space navigation, and surveillance. However, in order to be able to track an object undergoing 3-D motion using camera images one must recover depth, a missing dimension from a 2-D image. Hence, in order to retrieve the position of an object in the 3-D space a means to recover depth is necessary. In this study we assume that stereo vision [2] is used at first to enable the recovery of the depth from a sequence of "stereo" images. A problem associated with the use of stereo images is the matching of pixels from right and left images with the correct points on the object. In order to measure the depth of a point on a 3-D object, a point on the right image must be matched with a point on the left image screen. A matching algorithm which is a modification to the algorithm introduced in [5] was used for registration. Using the stereo camera images, the position and the velocity of an object were estimated using two different methods; first, by direct registration of the stereo images; and second, using an Extended Kalman Filter. Earlier work on the use of the Kalman Filter for object tracking includes that [4]. However, in [4], a single camera was used to estimate the position of an object undergoing pure translational motion with depth assumed to be constant and known.

The noise associated with the observations on the image screens has to be filtered out in order to achieve accurate estimates of the position and the velocity of the object. The transformation equations from 3-D to 2-D

[1] Research sponsored by SDIO/IST and managed by ONR, Contract N00014-86k-0515
[2] Currently with: Dept. of Elctr. Engr., Penn State Univ., Univ. Park, PA 16802

introduce nonlinearities in the observation model and thus the Extended Kalaman Filter (EKF) that allows for nonlinearities in the estimation model must be used. In order to improve the accuracy of the position estimates, the optic flow [3] was initially used along with the position on the image screen as additional measurement. The use of the optic flow, however, did not seem to improve the performance of the estimation. Consequently, we decided to omit the optic flow from our analysis. Instead, we decided to use an additional active sensor to improve the accuracy of the tracker. Thus, a range radar was used to estimate the object depth separately. The depth estimate was combined with the stereo camera images using an EKF to estimate the object position and velocity in the other directions.

2. ESTIMATION BASED ON DIRECT REGISTRATION OF STEREO IMAGES

2.1 The Matching Algorithm

Given the stereo camera setup, Fig. 1.1, with 2d the distance between the two cameras (assumed known), and f the cameras focal length, the transformation from a 3-D point with coordinates (x, y, z) to the left image point (x',y') and the right image point (x",y") is given by [2]

$$x' = \frac{f(x-d)}{f-z}, \qquad x'' = \frac{f(x+d)}{f-z}, \qquad y' = y'' = \frac{fy}{f-z} \tag{2.1}$$

From the right and left images the depth z can be recovered using (2.2)

$$z = f - \frac{2df}{x''-x'} \tag{2.2}$$

In order to recover the depth from (2.2), the pixels from the right and left images, Fig. 2.1, must be registered first correctly. In order to register the two stereo images, a point from the object must be matched with a point on each one of the two images. A matching algorithm, similar to the one introduced in [5], is used to find the most likely match between points on the right and left images. The algorithm is based on two assumptions: 1) each point in an image can only have one depth value; and 2) a point is very likely to have a depth value near the values of its neighbors. The slightly modified version of the algorithm [1] is given by

$$C_{n+1}(x,y,d) = \sigma \sum_{x',y',d' \in S} C_n(x',y',d') - \varepsilon \sum_{x',y',d' \in O} C_n(x',y',d') + \beta C_0(x',y',d') \tag{2.3}$$

where S corresponds to the excitatory region and O corresponds to the inhibitory region. The constants σ, ε, and β are arbitrary design parameters. The function C is given a value of one if a specified threshold is exceeded and a zero otherwise. The sigmoid

$$\mathrm{sigm}(x) = \frac{\exp(nx) - \exp(-nx)}{\exp(nx) + \exp(-nx)} \tag{2.4}$$

is used to smooth out the thresholded output. The excitatory and the inhibitory regions are illustrated in Figure 2.2. The eight excitatory points have the same depth as the point of interest. If some of the inhibitory points are on this will tend to keep the point of interest turned off, since only one depth value can be assigned to a point. Another important assumption in this matching algorithm is that both cameras are able to see the exact same part of the object. This means that there are no points on the object that are seen by only one of the two cameras.

2.2 Model of Translational Motion

In order to test the ability of the mathcing algorithm (2.3) to estimate the position of an object undergoing 3-D translational motion, a sequence of stereo images were generated using the model of a random accelerating object. The continuous-time dynamics of the object with random acceleration are described by the state equation

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0&1&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&0&0\\ 0&0&0&0&0&1\\ 0&0&0&0&0&0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0\\1\\0\\1\\0\\1 \end{bmatrix} w(t) \ , \text{ where } \quad \mathbf{x} = \begin{bmatrix} x\\ V_x\\ y\\ V_y\\ z\\ V_z \end{bmatrix} \tag{2.5}$$

is the state vector, and $w(t)$ is uncorrelated, zeromean, white, gaussian noise with covarariance $q(t)\delta(t-\tau)$, with $q(t) \geq 0$ for all t.

Notice that the dynamical model (2.5) is chosen to be unstable, constituting a worst case testing paradigm. Using (2.5) and the 3-D to 2-D projection equations (2.1) a sequence of images were generated, from which the position of the object was estimated using the matching algorithm (2.3).

2.4 Simulation

The model (2.5) was used to describe the 3-D motion of a flat thin surface that was used as the object in the simulation. The transformation equations (2.1), (2.2) were used to transform the position of the four corners of the object into pixels on the two image screens. All pixels on the two image screens located inside the four corner points were also turned on. The resulting two image screens were then fed into a matching algorithm (2.3) in order to match points on the two images. The matched pixels were then used to get an estimate of the depth of the object using (2.2).

The distance between the two cameras was set to be 8 meters so that the right and the left images were considerably different. The focal length, f, was 0.5 meters. The two cameras were assumed to be moving in order to be able to "see" the object at all times. The cameras move to the most recently estimated (x,y) location of the object between two consecutive images. The cameras are not moving in the z direction. Both images have a resolution of 16x16 pixels. The estimation errors in the x-direction are shown in Fig. 3.1. The estimate in the z direction (not shown) were clearly the most inaccurate. The main reason for the poor z estimate is the low resolution. The denominator of the z expression is especially affected by the resolution, since it depends on the difference between the two x estimates. Seeking improved position estimates, the extended Kalman Filter is considered next.

3. ESTIMATION BASED ON EXTENDED KALMAN FILTER AND STEREO CAMERA

The position and the velocity of the object are estimated given the observations of the location of the object on the two image screens. The observations are assumed to be noisy. The noise is introduced from inaccurate readings of the image screen as well as from low image resolution. The nonlinear transformation equations (2.1), (2.2) suggest the use of the Extended Kalman Filter (EKF) [7]. The dynamical model and the state vector are given by (2.9) and (2.10) respectively. The observation model for the EKF was obtained from the transformation equations (2.1), (2.2) by adding noise to account for the measurement noise at the camera and errors in the registration of the images. The EKF measurement vector is

$$z(t) = \begin{bmatrix} f(x(t)+d) \,/\, (f-z(t)) \\ fy(t) \,/\, (f-z(t)) \\ f(x(t)-d) \,/\, (f-z(t)) \\ fy(t) \,/\, (f-z(t)) \end{bmatrix} + v(t) \qquad (3.1)$$

where $v(t)$ is uncorrelated, zero mean, white gaussian, noise with covariance $r(t)\delta(t-\tau)$, with $r(t) \geq 0$ for all t. The initial conditions for the state vector are taken to be gaussian with mean $x(0)$ and positive definite covariance matrix $P(0)$. In (3.1), f is again the focal length and 2d the separation between the two cameras. Assuming constant acceleration during the sampling interval, the discrete time system is obtained from (2.5):

Process Model **Observation Model**

$$x_{k+1} = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x_k + w_k \qquad z_k = \begin{bmatrix} f(x_k+d) \,/\, (f-z_k) \\ fy_k \,/\, (f-z_k) \\ f(x_k-d) \,/\, (f-z_k) \\ fy_k \,/\, (f-z_k) \end{bmatrix} + v_k \qquad (3.2)$$

where T corresponds to the sampling time. The noise covariance matrices for w_k and v_k respectively are given by (3.3). For the EKF equations see [9].

$$Q = \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{3} & 0 & 0 & 0 & 0 \\ \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{T^4}{4} & \frac{T^3}{3} & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{T^4}{4} & \frac{T^3}{3} \\ 0 & 0 & 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \end{bmatrix} \qquad R = \begin{bmatrix} 1/T & 0 & 0 & 0 \\ 0 & 1/T & 0 & 0 \\ 0 & 0 & 1/T & 0 \\ 0 & 0 & 0 & 1/T \end{bmatrix} \qquad (3.3)$$

3.2 Simulation

Using the discrete time equations (3.2) through (3.5), the object position and velocity were estimated using the EKF, [1], [7], [9]. In order to prevent the object from moving out of the field of view of the stereo camera, the camera was assumed to track the object using the estimated velocity in the x,y directions. In the simulation, a focal length of 0.5 meter, a sampling time of 1.0 seconds, and a spacing between the two cameras of 0.1 meter were used. Assuming that the z-coordinate of the object was initailly -500 meters, the initial field of view is 100m wide, [9]. The fields of view of the two cameras are fairly narrow due to the large focal lengths. The sampling time of 1.0 second implies that images from the two cameras are available every second. A shorter sampling time will increase the performance of the filter, but since the processing of the images takes considerable computation time, a trade off has to be made. The sampling time is therefore set to be 1.0 second.

The observations are generated by the transformation equation from 3-D to 2-D using (3.1). It is assumed that the points from the right and the left image have been matched previously. The filter is run for 300 iterations and the state error along with the diagonal elements of the error covariance matrix , indicated as "camera model," are plotted and shown in Figures 4.1-4.5. The parameters q and r are constants that multiply the covariance matrices Q and R respectively. Note that the error in the velocity estimates is very small while the position error grows occasionaly before returning back to an acceptable range. The estimate in the z direction is the most inaccurate. This is due to the nonlinear transformation equations. The inaccuracy in z affects the other position components as well. The resulting estimation errors are fairly large and biased.

Since the z term introduces large errors in the estimation, the filter was run with fixed z and V_z in order to observe the difference in the estimation error. The resulting state errors and diagonal error covariance elements are shown in Fig.s 4.6-4.7. Notice how all the error covariance elements reach a specific value. The state errors are considerably smaller in this case. In addition, the state errors average out to zero.

The effect of the nonlinearities in the observation equation (3.2) can be studied by considering the Taylor's series expansion of the h vector in the EKF given by

$$\mathbf{h}(\mathbf{x}) = \mathbf{h}(\hat{\mathbf{x}}) + \mathbf{H}(\hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}}) + \text{H.O.T.} \tag{3.4}$$

where h and H have been defined previously and H.O.T. corresponds to higher order terms. The higher order terms are neglected in the filter. The approximation error that is made from neglecting the H.O.T. in (3.4) can subsequently be estimated. The nonlinearity in the observation equations (3.1) comes mainly from the z term in the denominator. Using (3.4), the nonlinearity in the denominator of the observation equations can be approximated by

$$\frac{1}{f - z} = \frac{1}{f - \hat{z}} + \frac{1}{(f - \hat{z})^2}(z - \hat{z}) + \text{error} \tag{3.5}$$

from which an approximate expression of the expected approximation error is obtained [1]

$$E\,[\text{error}] = E\left[\frac{(z-\hat{z})^2}{(f-z)(f-\hat{z})^2}\right] \approx \frac{P_{zz}}{(f-\hat{z})^3} \qquad (3.6)$$

(3.6) gives an expression for the error made in the approximation of $h(z)$ by the linear terms in (3.4). The error is plotted and shown in Fig. 4.8. The error is relatively small but introduces a bias on the state estimates.

4. DEPTH ESTIMATION THROUGH A RANGE RADAR

The model in section 3.1 produces inaccurate estimates of the object position and velocity. The estimation error in the z direction is especially inaccurate. It was seen in section 3.2 that the estimates can be greatly improved if the depth z were known precisely. The estimate obtained from the stereo camera could improve if accurate estimates of the depth z were available. A range radar is used to estimate the depth of the object separately. Once the depth is estimated, the estimate is fed to the camera filter to estimate the x, y components. The range radar is introduced in section 4.1 and the integration of the range radar filter and the camera filter is presented in section 4.2.

4.1 The Range Radar Filter

The range radar measures the distance (range) R to an object, along with two associated angles, the azimuth η, and the elevation ε [8], [9]. Using polar coordinates allows us to perform tracking in the system from which the measurements are obtained. The transformation between the polar coordinates (R,η,ε) and the Cartesian coordinate system (x,y,z) used in the camera model can be found in [8], [9]. The range radar filter is a coupled filter containing a range part along with an angle part. The angle filter consists of two individual filters for the two angles. The state vectors are given as follows

$$x_R = \begin{bmatrix} R \\ V_R \end{bmatrix} ; \quad x_H = \begin{bmatrix} \eta \\ V_H \end{bmatrix} ; \quad x_V = \begin{bmatrix} \varepsilon \\ V_V \end{bmatrix} \qquad (4.1)$$

The flow chart for the processing of this coupled filter is shown in Fig. 5.1. The system models are given by

$$x_R = \Phi_R x_R + w_R \quad ; \quad w_R \sim N(0,Q_R) \qquad (4.2a)$$

$$x_H = \Phi_H x_H + w_H \quad ; \quad w_H \sim N(0,Q_H) \qquad (4.2b)$$

$$x_V = \Phi_V x_V + w_V \quad ; \quad w_V \sim N(0,Q_V) \qquad (4.2c)$$

where the sampling index has been suppressed for simplicity. The measurement models are of the following form

$$z_R = h_R x_R + v_R \quad ; \quad v_R \sim N(0,R_R) \qquad (4.3a)$$

$$z_H = h_H x_H + v_H \quad ; \quad v_H \sim N(0,R_H) \qquad (4.3b)$$

$$z_V = h_V x_V + v_V \quad ; \quad v_V \sim N(0,R_V) \qquad (4.3c)$$

The transition matrices are defined as

$$\Phi_R = \begin{bmatrix} 1+\frac{wp^2 T^2}{2} & T \\ wp^2 T & 1+\frac{wp^2 T^2}{2} \end{bmatrix} \quad \Phi_H = \begin{bmatrix} 1 & \frac{T}{R_H} C_R \\ 0 & \rho_R \end{bmatrix} \quad \Phi_V = \begin{bmatrix} 1 & \frac{T}{R} C_R \\ 0 & \rho_R \end{bmatrix} \tag{4.4}$$

where

$$wp = \frac{\sqrt{V_H^2 + V_V^2}}{R} \qquad C = 1 - \frac{V_R T}{2R} \qquad R = R\cos\varepsilon \tag{4.5}$$

The observation matrices, h_R, h_H, h_V, are constructed based on that all the entries in the three state vectors are observable. The matrices are given by

$$h_R = h_H = h_V = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.8}$$

The error covariance matrices for the model and the observation noise have the following structure [8]:

$$Q_R = Q_H = Q_V = \frac{2\sigma_m^2}{\tau_m} \begin{bmatrix} \frac{T^5}{20} & \frac{T^4}{8} \\ \frac{T^4}{8} & \frac{T^3}{3} \end{bmatrix} \qquad R_R = R_H = R_V = \begin{bmatrix} \frac{1}{T} & 0 \\ 0 & \frac{1}{T} \end{bmatrix} \tag{4.9}$$

The linear Kalman filter [7] is used to estimate R, η, and ε. The transition matrices are updated of the beginning of each iteration. The estimates of R, η, and ε are used to generate the depth estimate according to $z = \hat{R}\cos\hat{\eta}\cos\hat{\varepsilon}$.

4.2 The Integrated Filter

The estimate of the depth obtained by the radar filter is used in the camera filter to help estimate the x and y coordinates. The integration of the two filters is illustrated in Fig. 5.1. It is assumed that the target motion can be accurately modeled as the motion of a randomly accelerating object. The actual data in the range radar filter is generated from the actual model through the transformation equations (4.3). However, in the range radar filter, it is assumed that the data is generated by a target undergoing a random maneuver during the interval between the 70th and the 90th time step. Thus, an intentional mismatch between the actual model and the perceived range radar model is introduced to test the robustness of the range radar filter and the integrated filter, [9]. The estimate of the depth is used in the transformation equations in the camera filter where it is treated as a constant. Thus, the resulting Kalman filter is linear. The cameras are moving as described in section 2.4. The object motion is strictly translational. The rotational motion is covered in section 4.5.

4.3 Simulation

The integrated filter in the previous section was simulated with the following parameters: T=1.0, f=0.5, d=0.1, q=0.01 (for camera filter),q=0.1

(for range filter), $r = 0.0068$ (for range radar filter), $r = 0.01$ (for camera filter and angle filters), $\tau_m = 100$ (maneuver time constant in range radar), $\sigma_m = 1.0$ (maneuver standard deviation).

The choice of a lower r for the range radar is based on the assumption that observations in this case are fairly accurate. The range radar filter assumes that the object maneuvers in the interval between 70 and 90 iterations. The parameters that are associated with this maneuvering is given above. The resulting estimate errors and the related error covariance elements are shown in Fig.s 4.1-4.5. Comparing these figures to the figures in section 3 it is easily seen that the errors are reduced. The errors average to zero as in the fixed z case in section 3. The elements of the error covariance matrices behave better as well. The error covariance elements for the range radar are reinitialized when the difference between an element in two consecutive iterations is smaller that 0.001. Note how the errors are decreased every time a reinitializing occurs.

4.4 Estimation Based on Mono Camera

Since the depth in the integral filter is estimated with measurements from the range radar, the use of the stereo camera seems redundant. Comparison of the x- direction estimates, similarly in the other directions, obtained with a mono camera, Fig.s 5.3-5.4, with their stereo camera counterparts, indicates that the estimation errors and the error covariances are higher in the mono camera case. The use of a stereo camera is therefore justified.

4.5 Rotational Object Motion

The previous models have assumed that the object moves with only translational motion. Naturally an object very rarely moves with zero rotational velocity. In this section rotational object motion is introduced.

Initially the rotational velocity is assumed to be known and constant. The rotation is taken into account in a modified model of (2.5). The observation equations remain the same. The z and zvel estimates are fed into the camera filter from the range radar and will be treated as inputs in the camera model. The resulting discrete model is then given by

$$x_{k+1} = \begin{bmatrix} 1 & 0 & T & -\omega_z T & 0 & 0 \\ 0 & 1 & 0 & 0 & -\omega_z T & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \omega_z T & 0 & 0 & 1 & 0 & T \\ 0 & \omega_z T & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x_k + \begin{bmatrix} \omega_y T & 0 \\ 0 & \omega_y T \\ 0 & 0 \\ -\omega_x T & 0 \\ 0 & -\omega_x T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_k \\ Vz_k \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Wx_k \\ Wy_k \end{bmatrix} \qquad (4.10)$$

where $(\omega_x, \omega_y, \omega_z)$ are the known constant angular velocity. The covariance matrix Q_w of the noise W_k is given by

$$Q_w = \begin{bmatrix} q\Delta T & 0 \\ 0 & q\Delta T \end{bmatrix} \qquad (4.11)$$

The state vector $\mathbf{x}$ is given by

$$\mathbf{x} = \begin{bmatrix} x \\ V_x \\ V_{t,x} \\ y \\ V_y \\ V_{t,y} \end{bmatrix} \tag{4.12}$$

The covariance Q that is used in the filter equations is given by

$$Q = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} Q_w \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & q\Delta T & q\Delta T & 0 & 0 & 0 \\ 0 & q\Delta T & q\Delta T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & q\Delta T & q\Delta T \\ 0 & 0 & 0 & 0 & q\Delta T & q\Delta T \end{bmatrix} \tag{4.13}$$

The range radar model is the same as before since it already incorporates constant angular velocities. The above model was simulated with essentially the same parameters as in the translational case. The sampling time was 1.0 second and q was set to 0.01. The angular velocities were all set to 0.011 rad/sec. The resulting estimation errors and the corresponding error covariances are shown in Fig.s 5.5-5.6. The estimation errors in the position are basically the same as they were for the purely translational case, whereas the velocity estimates are worse.

Next we consider the case of random angular acceleration. The angular velocities cannot be treated as constants in this case. Both the camera filter and the range radar filter have to be modified. In order to avoid additional nonlinearities in the camera filter, the angular velocities are estimated in the range radar and fed into the camera filter just like the estimates for z and zvel are. The augmented state vectors in the range radar are given by

$$\mathbf{x}_R = \begin{bmatrix} R \\ V_R \\ \omega_R \\ \dot{\omega}_R \end{bmatrix} ; \quad \mathbf{x}_H = \begin{bmatrix} \eta \\ V_H \\ \omega_H \\ \dot{\omega}_H \end{bmatrix} ; \quad \mathbf{x}_V = \begin{bmatrix} \varepsilon \\ V_V \\ \omega_V \\ \dot{\omega}_V \end{bmatrix} \tag{4.14}$$

where $(\omega_R, \omega_H, \omega_V)$ is the angular velocity. The system models are given by

$$\mathbf{x}_R = \Phi_R \mathbf{x}_R + \mathbf{w}_R \qquad ; \ \mathbf{w}_R \sim N(0, Q_R) \tag{4.15a}$$

$$\mathbf{x}_H = \Phi_H \mathbf{x}_H + \dot{\omega}_V\, RT + \mathbf{w}_H \qquad ; \ \mathbf{w}_H \sim N(0, Q_H) \tag{4.15b}$$

$$\mathbf{x}_V = \Phi_V \mathbf{x}_V - \dot{\omega}_H\, RT + \mathbf{w}_V \qquad ; \ \mathbf{w}_V \sim N(0, Q_V) \tag{4.15c}$$

The state transition matrices are defined by

$$
\Phi_R = \begin{bmatrix} 1 + \frac{wp^2 T}{2} & T & 0 & 0 \\ wp^2 T & 1 + \frac{wp^2 T}{2} & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (4.16a)
$$

$$
\Phi_H = \begin{bmatrix} 1 & \frac{T}{R_H} C_R & 0 & 0 \\ 0 & \rho_R & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \Phi_V = \begin{bmatrix} 1 & \frac{T}{R} C_R & 0 & 0 \\ 0 & \rho_R & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (4.16b)
$$

where all the parameters have been defined previously. The error covariance matrices for the model noise have the following structure

$$
Q_R = Q_H = Q_V = \frac{2\sigma_m^2}{\tau_m} \begin{bmatrix} \frac{T}{20} & \frac{T}{8} & 0 & 0 \\ \frac{T^4}{8} & \frac{T^3}{3} & 0 & 0 \\ 0 & 0 & \frac{T^3}{4} & \frac{T^3}{3} \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \end{bmatrix} \qquad (4.17)
$$

The camera model is modified in the following way

$$
x_{k+1} = \begin{bmatrix} 1 & 0 & T & -\omega_z T & 0 & 0 \\ 0 & 1 & 0 & -\dot{\omega}_z T - \omega_z T & & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \omega_z T & 0 & 0 & 1 & 0 & T \\ \dot{\omega}_z T & \omega_z T & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x_k + \begin{bmatrix} \omega_y T & 0 \\ \dot{\omega}_y & \omega_y T \\ 0 & 0 \\ -\omega_x T & 0 \\ -\dot{\omega}_x T - \omega_x T & \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_k \\ Vz_k \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Wx_k \\ Wy_k \end{bmatrix} \qquad (4.18)
$$

where $(\omega_x, \omega_y, \omega_z)$ and $(\dot{\omega}_x, \dot{\omega}_y, \dot{\omega}_z)$ are estimated in the range radar filter with the use of the transformations in Appendix B in [1], [9].

The model described above was simulated. The parameters in (4.11) were used. The initial values for the rotational state vector entries were selected as follows:

$$
(\omega_R, \omega_H, \omega_V) = (0.01, 0.01, 0.01)\,(\dot{\omega}_R, \dot{\omega}_H, \dot{\omega}_V) = (0.001, 0.001, 0.001) \qquad (4.19)
$$

The resulting estimation errors and the corresponding error covariances for the object position in the x-direction are shown in Fig.s 5.7-5.8. The errors are close to previous results. Figures of the estimates in the other directions and in the associated velocities can be found in [1], and [9]. The overall performance of the filter degrades when the angular velocity changes randomly as expected.

CONCLUSION

Three different approaches for estimating the position of and tracking an object undergoing 3-D transaltional and rotational motion were considered. One approach involved a stereo camera and position estimation directly from stereo image registration. The second approach involved a stereo camera and use of an Extended Kalman Filter (EKF) for position and velocity estimation. In the third approach, a range radar was used to estimate the depth from separate measurements. The depth estimate was subsequently used in an EKF to recover the object position and velocity (both translational and angular) from a sequence of stereo images. Numerical comparison of the three approaches via simulation indicates that the range radar - EKF integral filter is superior to the other two approaches, Fig.s 4.1-4.5. Furthermore, the integral filter can track successfully objects undergoing 3-D translational and rotational motion. From the simulation results is seen that the effects of random rotation are more visible in the velocity estimates [1], [9]. The position estimates were very closed to those obtained in the purely translational motion case. The performance is, therefore, not affected by the random angular accelaration, except for the estimates of the compound velocities.

REFERENCES

[1] Nilsson, L., "An Integrated Tracking System Using a Stereo Camera and a Range Radar", Master's Thesis, Southern Illinois University, 1989.

[2] Horn, B. K. P., "Robot Vision", Cambridge, Ma:MIT Press, 1986.

[3] Kanatani, K., "Transformation of Optical Flow by Camera Rotation", IEEE Transactions on Pattern and Machine Inelligence, Vol. 10, No. 2, March 1988, pp.131-143.

[4] Broida, T. J., and Chellappa, R., "Estimation of Object Motion Parameters from Noisy Images", IEEE Transactions on Pattern and Machine Intelligence, Vol. 8, No.1, Jan. 1986, pp. 90-99.

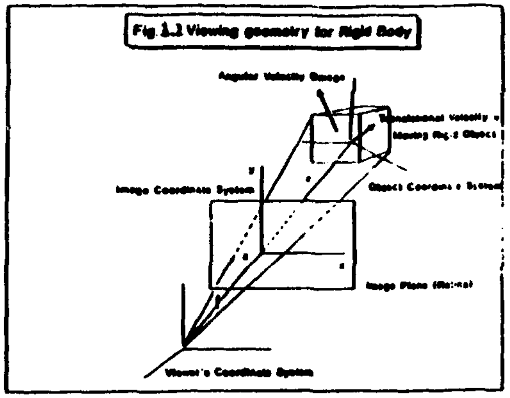
[5] Marr D., and Poggio, T., "Cooperative Computation of Stereo Dispariaty", Science 194, 1976, pp. 283-287.

[6] Ballard, D. H., and Kimball, O. A., "Rigid Body Motion from Depth and Optical Flow", Computer Vision, Graphics, and Image Processing 22, 1983, pp. 95-115.

[7] Gelb, A., editor, "Applied Optimal Estimation", Cambridge, MA, MIT Press, 1974.

[8] Blackman, S. S., "Multiple Target Tracking with Radar Applications", Artech House INC., Norwood, Ma, 1986.

[9] Thomopoulos, S.C.A., and Nillson, L., "Object Tracking from Image Sequences Using Stereo Camera and Range Radar," Tech. Report, TR-SIU-DEE-89-3, IPIS Lab, Electr. Engr. Dept., Southern Illinois Univ., Carbondale, IL 62901.



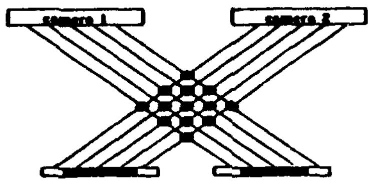


Figure 2.1 Merging of z-values from the two images

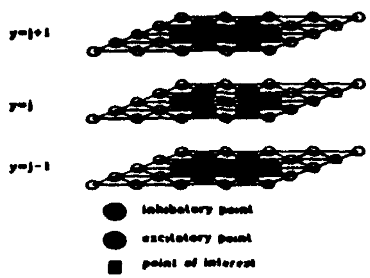


Figure 2.2 Excitatory and inhibitory regions

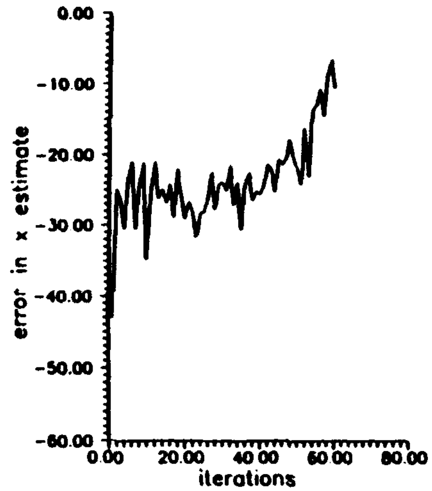


Figure 3.1 Error in x-estimate based on images alone; no filter used

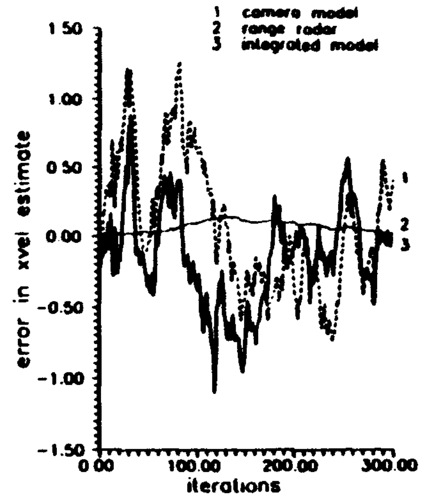


Figure 4.2 Error in xvel-estimate using camera model, integrated model and range radar model

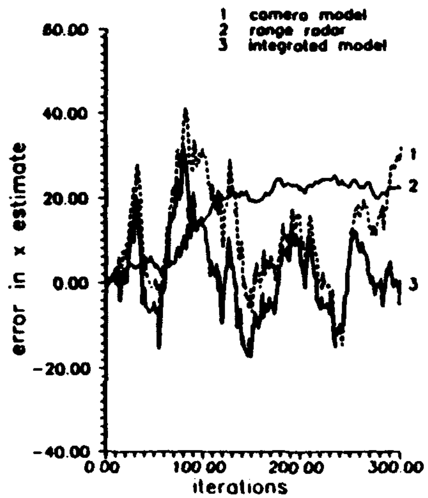


Figure 4.1 Error in x-estimate using camera model, integrated model and range radar model

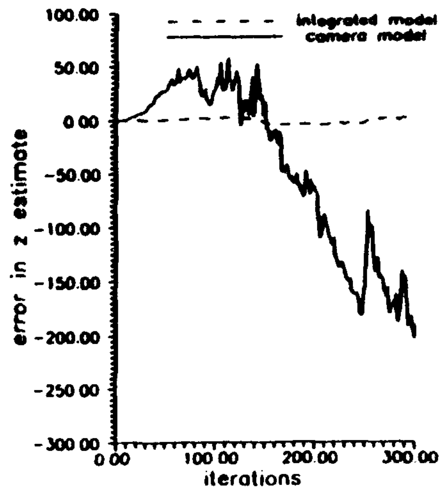


Figure 4.3 Error in z-estimate using camera model and integrated model

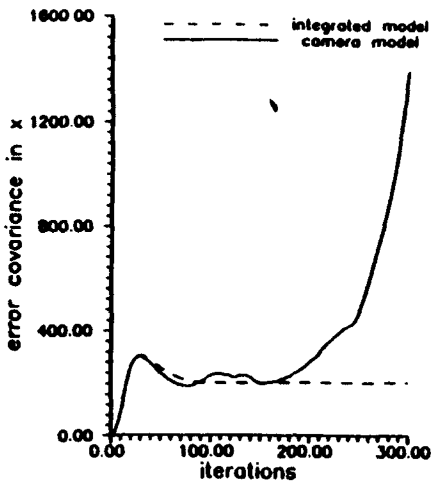


Figure 4.4 Error covariance in x using camera model and integrated model

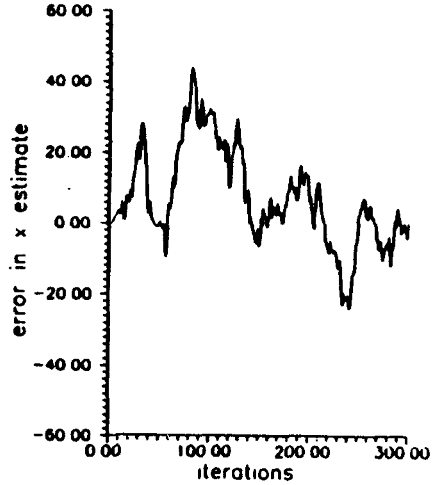


Figure 4.6 Error in x-estimate using the camera model with known z values

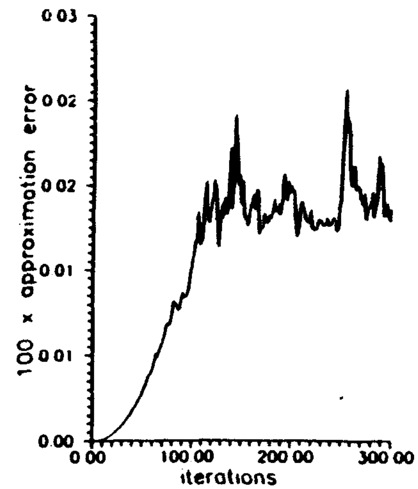


Figure 4.8 Error due to the Taylor series expansion for camera model

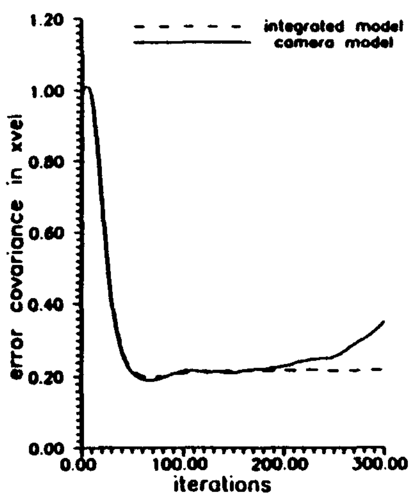


Figure 4.5 Error covariance in xvel using camera model and integrated model

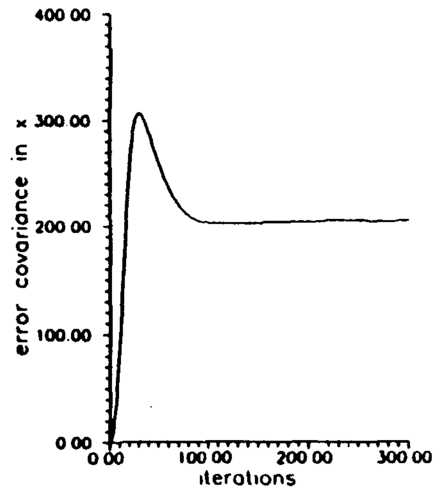


Figure 4.7 Error covariance in x using the camera model with known z values

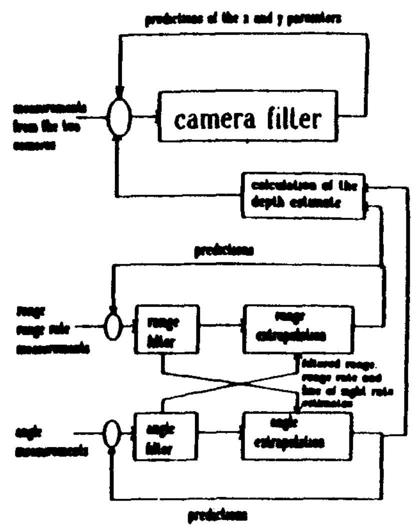


Figure 3.2 Processing flow chart for the integrated filter

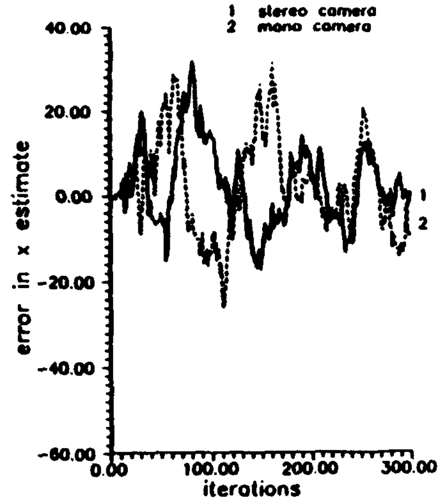


Figure 5.3 Error in x-estimate using a mono camera and a stereo camera; the integrated model is used

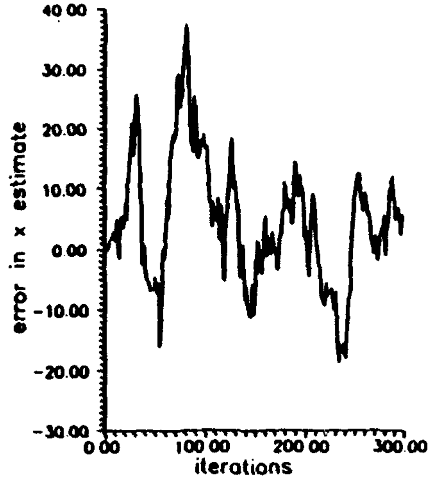


Figure 5.5 Error in x-estimate with known constant rotation using the integrated model

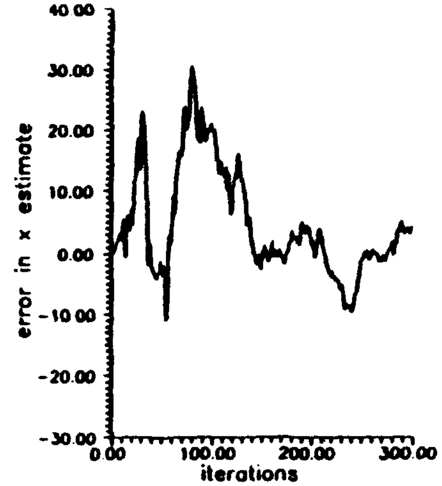


Figure 5.7 Error in x-estimate in the presence of rotation using the integrated model

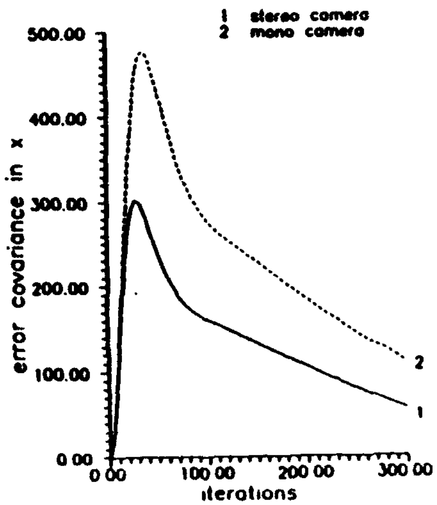


Figure 5.4 Error covariance in x using a mono camera and a stereo camera, the integrated model is used

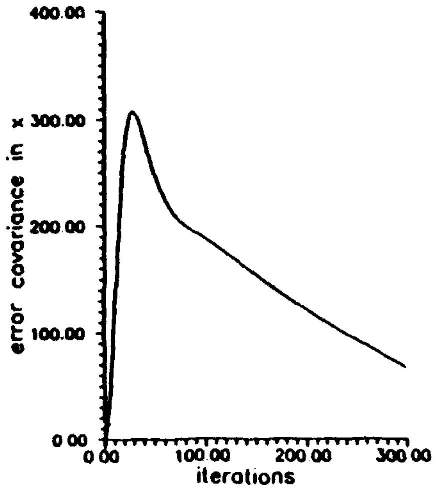


Figure 5.6 Error covariance in x with known constant rotation using the integrated model

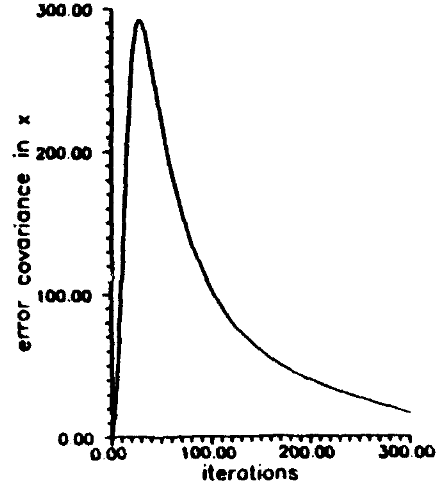


Figure 5.8 Error covariance in x in the presence of rotation using the integrated model

DIGNET: A Self-Organizing Neural Network for Automatic Pattern Recognition, Classification and Data Fusion

Stelios C. A. Thomopoulos *

Decision and Control Systems Laboratory

Department of Electrical & Computer Engineering

The Pennsylvania State University

University Park, PA 16802

Dimitrios K. Bougoulias †

Department of Electrical Engineering

University of Southern Illinois

Carbondale, IL 62901

Abstract

DIGNET is a self-organizing artificial neural network (ANN) that exhibits deterministically reliable behavior to noise interference, when the noise does not exceed a pre-specified level of tolerance. The complexity of the proposed ANN, in terms of neuron requirements versus stored patterns, increases linearly with the number of stored patterns and their dimensionality. The self-organization of the DIGNET is based on the idea of competitive generation and elimination of *attraction wells* in the pattern space. DIGNET is used for Pattern Recognition and Classification and for Signal Detection and Fusion. Analytical and numerical results are included.

1 Introduction

Most artificial NN's (ANN's) that are used in the literature for pattern recognition and classification require that the patterns that are stored and recognized be orthogonal with each other ([1], [2], [3], [4], [5], [6], [7]). Furthermore, they are usually vulnerable to noise interference, in the sense that a usually small deviation from the orthogonality assumption renders them unstable. For a viable neural-based solution to the recognition/classification problem in the presence of noise, the artificial neural network must be designed so that it is, by design and not by incident, robust to prespecified noise margins. *DIGNET*, the artificial neural network that we propose for automatic pattern recognition and classification, signal detection and distributed data fusion, reflects this philosophy.

2 Proposed Artificial Neural Network Architecture

Ideally, the input-output characteristic of an ANN that is used for pattern recognition and classification in cluttered noise should resemble that of Fig. 1. In Fig. 1, the horizontal curves represent "attraction wells" around the stored patterns. If the stored patterns are identified with equilibrium points of the ANN dynamics, then the attraction wells of Fig. 1 represent attraction regions around these points in a multidimensional space. Thus, if the noise is identified as a percentage disturbance of the stored patterns, the attraction wells represent hyperspheres of predetermined radius around the patterns. So, if the ANN is presented with a distorted pattern that lies in one of these attraction regions, correct recognition (and classification) will be guaranteed from the convergence of the ANN to the correct equilibrium point. If, on the other hand, the ANN is initially presented with a pattern that lies outside any of the attraction regions, a new attraction well will be created and the ANN will converge to the *unknown pattern* as it should. Thus, an ANN with the characteristic of Fig. 1 exhibits learning capabilities, since new patterns can be stored by extending the attraction points of the operating characteristic in Fig. 1. Furthermore, the noise tolerance of the ANN can be changed by modifying the "width" of the attraction wells. Dignet dynamically realizes the ideal characteristic of Fig. 1.

3 Directors and the unity hypersphere

In linear system theory eigenvectors have only meaning as directions, their magnitude being undetermined. Any vector that lies in the direction of an eigenvector of the system is also an eigenvector independent of its magnitude.

*This research was partially supported by SDIOT/IST and managed by the office of Naval Research under contract N00014-K-0515.

†Presently with the Greek Air Force and with QUALITY Imports/Exports, Inc., Athens, Greece.

0-8194-0748-8/92/$4.00

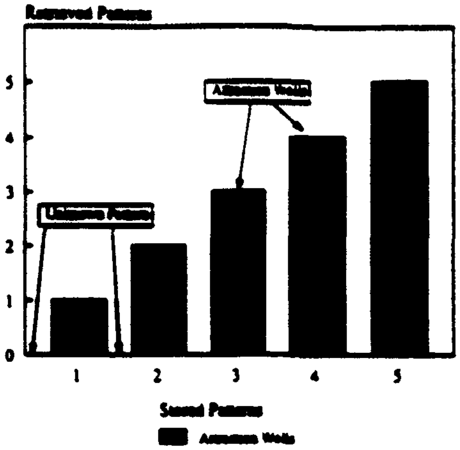


Figure 1: Ideal characteristic of ANN for Automatic Pattern Recognition and Classification

On the other hand, a pattern is well defined irrespective of scaling or reversal. For instance we can recognize a visual shape even under different light intensities (scaling), even if we see the photographic negative (reversal). The above examples can motivate the conceptualization of patterns as straight lines in the n-dimensional space. To further understand the operation of Dignet we introduce a mathematical entity that we call "*director*."

Definition 3.1 *An n-dimensional director is the set of all vectors lying on the same straight line passing through the origin of an n-dimensional vector space. We use the notation* $a, b, c, d \ldots$ *to indicate directors.*

We shall prove that the set of all n-dimensional directors (n-directors) is a metric space.

Definition 3.2 *For two n-directors* a, b *we define as distance* $\Theta(a, b)$ *the absolute value of the acute angle between any two of their elements. In terms of the vector space it can be expressed as*

$$\Theta(a, b) = \arccos\left(\left|\frac{< x, y >}{\|x\|\|y\|}\right|\right) \tag{1}$$

where x, y *vectors so that* $x \in a$, $y \in b$.

It is easy to see that this distance fulfills all the properties of a metric:

1. Nonnegative because $\arccos(x) \in [0, \pi/2]$ for $x \in [0, 1]$ (CBS inequality)
2. Symmetric, obviously if we interchange a and b in the formula.
3. The triangle inequality clearly holds for the 3-dimensional space (with equality when all 3 directors lie on the same plane). However, any three, non-collinear vectors (or straight lines) span a 3-dimensional subspace in the n-space that is homomorphic to the 3-D space. Therefore, the metric properties hold for the n-dimensional space, too.

Thus, the set of all n-directors is a metric space.

From the definition 3.1 it follows that a director being a set can be represented by one of its elements. A good choice is the *unity vector* that belongs to the particular director. This choice simplifies equation 1. If X and Y are unity vectors representing the directors a and b respectively, then

$$\Theta(X, Y) = \arccos(| < X, Y > |) \tag{2}$$

and the directors can be further represented as points on the surface of the unity *hypersphere*. In figure 2 we see the 3-dimensional case. This mapping of pattern vectors to unity vectors can be achieved by normalization and reduces a n-dimensional problem to a $(n-1)$-dimensional problem.

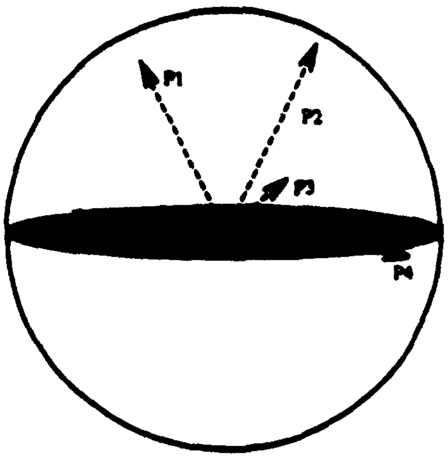


Figure 2: Normalized patterns and the unity sphere

The topological properties of this mapping are interesting, however the algebraic properties are complicated. Therefore, to simplify things we assume that the angles are small, then in the limit the surface of the sphere can be treated as a tangent plane. Then if we consider a neighborhood $N(P_i, \theta)$ around the pattern P_i, where θ (theta) is the desired angular threshold for pattern matching, we say that a pattern is recognized by the exemplar P_i if its projection on the surface of the sphere falls within the above neighborhood.

Since the vectors are already normalized, the angle corresponds to the inner product between vectors, and the comparison of a new pattern with a number of prestored exemplars can be achieved with a simple parallel vector matrix multiplication and thresholding of the output, where the rows of the matrix correspond to the exemplars.

4 Description of Dignet

Dignet is a self-organizing neural network that can store and classify noisy inputs without supervised training. Its self-organization capability is based on the idea of competitive generation and elimination of attraction wells. The wells are generated around presented patterns which are clustered according to their distance from the center of wells. The center of a well is moving dynamically towards the highest concentration of clustered points in the pattern constellation. The depth of a well indicates the strength of learning and reflects into the inertia by which the center of the well is moving when new data falls within its region of attraction.

A schematic diagram of Dignet is shown in Fig. 3. The pattern recognition and classification ability of Dignet is characterized by the competitive creation and elimination of attraction wells. Each well is characterized by its center, width (threshold), and depth. The similarity between patterns in Dignet is measured in terms of the angle that the patterns form among themselves. It is assumed that all patterns are normalized, so that the magnitude of a pattern does not affect the classification capability of the network. Assuming that a number of wells has already been created, the changes in the Dignet geography once a new pattern is presented are as follows.

Let x_n represent the pattern that is presented to Dignet at the n-th time instant. If e_{n-1} represents the center of an existing well in Dignet at the time the new pattern is presented, the center changes according to

$$e_n = \frac{c_n}{d_n} x_n + \frac{d_{n-1}}{d_n} e_{n-1}, \text{ with initial conditions } e_0 = 0. \tag{3}$$

where d_{n-1} is the depth of the well at the $n-1$st presentation, which is updated according to

$$d_n = d_{n-1} + c_n, \text{ with initial conditions } d_0 = 0. \tag{4}$$

and c_n is a variable that takes on the following values

$$c_n = \begin{cases} 1 & \text{if the pattern is won by the well (reinforcement)} \\ 0 & \text{if the pattern does not fall in the well (no interaction)} \\ -1 & \text{if the pattern falls in the well, but is not won by it} \end{cases} \tag{5}$$

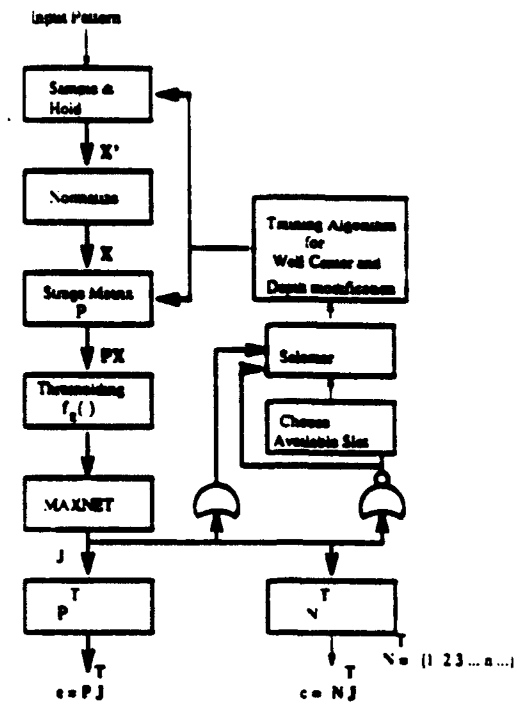


Figure 3: Schematic diagram of DIGNET

The width of a well (threshold) determines the region of attraction and is determined by the specified (desired) signal-to-noise ratio (SNR). The threshold is measured in degrees of angle from the center of the well. Given a SNR, the threshold (cosine) is determined by

$$\text{threshold} = \sqrt{\frac{1}{1 + 10^{-\frac{SNR}{10}}}} \tag{6}$$

and the well width (in degrees)

$$\Theta_0 = \arccos(\text{threshold}) \tag{7}$$

Equation 6 is obtained from figure 4. The noise component that contributes to the angular deviation from the center of the well s is normal to the pattern. Therefore for worst case analysis we can assume that the noise is normal to the pattern. If we cut the n-dimensional hypersphere by a 2-dimensional plane so that the vector s lies on that plane as well as the center of the hypersphere, then we reduce the problem to an equivalent 2-dimensional problem. Then $< n, n > +1 = \lambda^2 =< s + n, s + n >$ by the Pythagorean theorem. Then

$$\cos(\Theta) = 1/\lambda = \frac{1}{\sqrt{1 + < n, n >}} \tag{8}$$

By using $< n, n >= \sigma^2$ (in expected value sense), we obtain

$$\cos(\Theta) = \frac{1}{\sqrt{1 + \sigma^2}} \tag{9}$$

from which, using the relation:

$$\sigma^2 = 10^{-\frac{SNR}{10}} \tag{10}$$

(6) follows.

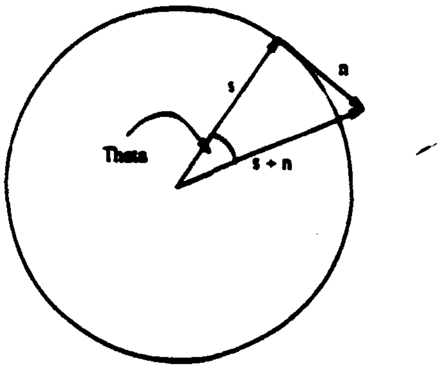


Figure 4: 2-dimensional projection of pattern and normal noise

Once a pattern is presented to the network, its distance from the different wells is computed. If the minimum distance exceeds the well width, a new well is created: otherwise the pattern is assigned to the closest well, which is reinforced. Furthermore, if the pattern, in addition to falling in the region of attraction of its closest well, falls in the region of attraction of other wells as well, these wells are weakened, their center is pushed away and their depth decreases according to the above equations. To avoid excessive, spurious wells, a *stage age* (s.a.) is defined. The depth of each well is periodically examined at the end of each s.a.. If at the end of a s.a., the depth of a well does not exceed a certain threshold (age), the well is eliminated all together: otherwise it survives this stage age.

5 Stability and Convergence Analysis of Dignet

For reasons of analytical compactness, we perform a stability and convergence analysis of Dignet by using the continuous time equivalent of the self-organizing algorithm (equations 3 through 7). Simple manipulation of the discrete-time algorithm, yields the following continuous time algorithm:

$$\frac{d}{dt}[e_i(t)d_i(t)] = \frac{d}{dt}[d_i(t)]x(t) \tag{11}$$

$$\frac{d}{dt}[d_i(t)] = I[\Theta_0 - \Theta(e_i(t), x(t)) > 0](2I[\Theta(e_i(t), x(t)) = \min_j\{\Theta(e_j(t), x(t))\}] - 1) \tag{12}$$

where $e_i(t)$ designates the center of the i-th well in Dignet at time t, $d_i(t)$ the associated depth, and

$$\Theta(e_i, x(t)) = \arccos\left(\left|\frac{< e_i(t), x(t) >}{||e_i(t)||||x(t)||}\right|\right) \tag{13}$$

$I[\Omega]$ is the indicator function defined to be one if the event Ω is true, and zero otherwise. The minimum in 12 is understood over all existing wells in Dignet at time t.

Assuming zero initial conditions on $d(0)$, i.e. $d(0) = 0$, the solution to the differential equation 11 is

$$e(t)d(t) = \int_0^t \dot{d}(\tau)x(\tau)d\tau \tag{14}$$

where the notation "$\dot{d}(t)$" is used to indicate the time-derivative of $d(t)$. Assuming $d(t) \neq 0$, and using the convention $\frac{0}{0} := 0$, the solution 14 can be written as

$$e_i(t) = \frac{\int_0^t \dot{d}_i(\tau)x(\tau)d\tau}{d(t)} = \frac{\int_0^t \dot{d}_i(\tau)x(\tau)d\tau}{\int_0^t \dot{d}_i(\tau)d\tau} \tag{15}$$

For the i-th Dignet well with center $e_i(t)$, the integral in the denominator of (15) represents the average time that any input pattern $x(t)$ fell into the region of attraction of well i and won by this well (i.e., it was closest to the center

$e_i(t)$ than to any other well center), minus the average time that any other pattern fell into the region of attraction of well i, but was lost over to competition. (The convention $\frac{0}{0} := 0$, is assumed in the analysis.) Thus, (15) produces wells with centers the *selective* time-averages of different input noisy patterns. Furthermore, it eliminates wells that are created from overlapping well boundaries. The solutions (15) are stable, assuming finite mean data, and converge to either a time average if the pattern persists in the input data, or zero if the pattern is spurious. The storage parameter, s.a., that was introduced in the description of Dignet facilitates the elimination of unsustained and undesired spurious wells, in order to keep the storage capacity requirements of Dignet manageable. The algorithm (equations 11 through 13) or, its equivalent discrete time version (equations 3 through 7), is thus capable of self-organization and can be used in a neural network for class-discrimination among different classes that are separable by hyperspheres. Classes of patterns which are separable by more complicated boundary shapes can be discriminated by Dignet through self-organization, if a different metric is used to determine the interaction among input patterns and well centers, other than the angle metric (1) used in the indicator function $I[\{\Theta_0 - \Theta(e_i(t), x(t)) > 0\}]$ in (12) [8]

6 Comparison with other self-organizing networks

Kohonen [9] has proposed a class of self-organizing feature maps that are based on the adaptation law

$$\frac{m(t)}{dt} = o(x, m, \eta)x(t) - \gamma(x, m, \eta)m(t) \tag{16}$$

$$\eta(t) = m^T(t)x(t) \tag{17}$$

where $\eta(t)$ represents the neuron activation (or output for linear elements), $x(t)$ is the vector of the input excitations to the neuron, and $m(t)$ is the vector of the synaptic interconnections associated with the neuron and the input vector $x(t)$. $o(\cdot)$ and $\gamma(\cdot)$ are, in general, functions (possibly nonlinear) of the synaptic weights m, the input x, and the neuron output η. In Kohonen's self-organization feature maps [9], the class of functions $o(\cdot)$ and $\gamma(\cdot)$ that he considers are memoryless functions.

To compare DIGNET with Kohonen's maps we rewrite equations (11) and (12), by dropping the time-dependence for notational convenience, as follows:

$$c = -\frac{d}{d}e + \frac{d}{d}x \tag{18}$$

$$d = I[\Theta_0 - \Theta(e_i(t), x(t)) > 0](2I[\Theta(e_i(t), x(t)) = \min_j\{\Theta(e_j(t), x(t))\}] - 1) \tag{19}$$

with output equation

$$r = \max_i\{Pe_i\} \tag{20}$$

where the maximum is taken over all center-patterns of created wells (clusters), and P is the matrix of the stored patterns (a matrix with the stored patterns as rows). By comparing equation (16) with equations (18) and (19), and identifying

$$o = \gamma = \frac{d}{d}e \tag{21}$$

the Dignet algorithm extends the class of Kohonen's feature maps by introducing memory in $o(\cdot)$ and $\gamma(\cdot)$. Another class of algorithms that can learn to discriminate among a number of different patterns (hypotheses), are based on the learning vector quantization (LVQ) algorithm and the creation of Voronoi vectors in the pattern space [10], [11]. However, the LVQ algorithm, and derivative algorithms from it, requires that the number of unknown patterns (hypotheses) is precisely known a priori, much the same way Kohonen's self-organizing feature maps do. Furthermore the number of Voronoi vectors must be close to the true number of different clusters in the pattern space. For convergence, the LVQ algorithm must be initialized with the proper number of Voronoi vectors and initial conditions that are close to the stable equilibrium points. A modification of the LVQ algorithm that allows the adaptive update of the Voronoi vectors according to a majority decision rule was proposed in [11]. The modified LVQ algorithm avoids the instability of the original LVQ algorithm due to bad initial conditions, but it requires that the size of the Voronoi cells remains small, thus, not really resolving the sensitivity problem of the algorithm.

If the initial choice of the Voronoi vectors in the LVQ algorithm is inadequate, there is no systematic approach to adaptively change their number as needed. Convergence of the LVQ algorithm depends on the proper choice of the

Voronoi vectors and initialization of the algorithm close to the actual stable point. Convergence of the Kohonen's feature maps depends on the choice of $\sigma(\cdot)$ and $\gamma(\cdot)$ functions, which are otherwise arbitrary. In that sense, neither the LVQ algorithm nor Kohonen's feature maps are truly self-organizing in the sense defined by Dignet, since the number of different patterns need to be known a-priori, and convergence is sensitive to the choice of initial conditions. In that respect, the guaranteed convergence of the Dignet algorithm to a number of stable classes, given noisy data from an unknown number of unknown patterns represents the novelty of the algorithm that differentiates if from the LVQ algorithm and Kohonen's feature maps.

7 Capacity of Dignet

Determination of the maximum capacity on Dignet to store patterns unambiguously depends on the metric that is used in the well formation, the dimensionality of the patterns, and their separation from each other in the absence of noise. The maximum capacity of Dignet to store input patterns unambiguously depends on the maximum amount of tolerable deformation, which depends on the prescribed SNR, and the initial separation of the patterns. The capacity of Dignet when the metric (1) is used in the well formation is discussed next.

For n-dimensional input patterns, assuming that the separation between patterns is equal to $\Theta_0 = \arccos(\text{thresh})$, where thresh $= (1+\sigma^2)^{-\frac{1}{2}}$ with $\sigma^2 = 10^{-SNR/20}$ the noise variance and Θ_0 is measured in radians, an approximation of the maximum capacity of Dignet is given by

$$C_n \approx \left(\frac{\pi}{2\Theta_0}\right)^{n-1} \tag{22}$$

if a pattern and its negative are indistinguishable, and by

$$C_n \approx \left(\frac{\pi}{\Theta_0}\right)^{n-1} \tag{23}$$

when a pattern is distinct from its negative.

The maximum number of unambiguous classes that Dignet can create increases within the dimensionality of stored patterns, since the number is proportional to ratio of the surface of the hypersphere where the well centers are situated to the surface occupied by the width of a well. The estimates on the maximum capacity of Dignet are thus obtained by comparing the area of the surface of the n-dimensional sphere with the area of the hyperdome of solid angle Θ_0. Notice that this capacity can be much higher than the capacity of conventional neural networks and it is limited only by the minimum desired distance between exemplars that is dictated by the amount of noise that the network is required to be able to tolerate. The advantage of Dignet lies, thus, on its ability to create classes with prespecified noise tolerance. For example, for tolerance to SNR = 0 db, $\sigma^2 = 1$, thresh $= 2^{-\frac{1}{2}}$ which corresponds to $\Theta_0 = \pi/4$, and thus $C_n = 2^{n-1}$ for indistinguishable negative from positive patterns, and 4^{n-1} for distinct positive from negative patterns. Hence, for 0 db SNR, the well width should be set at 90^0 which corresponds to a threshold of 45^0. For tolerance to SNR = 24db, the well width drops to 26^0, which corresponds to threshold of only 13^0, which yields a lower bound on the maximum capacity of Dignet equal to 6.67^{n-1} or 13.34^{n-1} depending on whether the Dignet is designed to be insensitive to orientation of not.

8 Implementation of Dignet

An implementation of Dignet is shown schematically in Fig.3. The different input patterns are represented by vectors that are stored directly as rows of the matrix P. The vectors are first normalized to render the recognition and classification abilities of the network insensitive to magnitude variations in input patterns. Since Dignet may be used for recognition and classification, the network must be independent of the relative level of intensity in the input patterns. Normalization of the input patterns creates equivalence classes between collinear patterns.

Once an input pattern is presented in Dignet, it is first sampled, and the samples vector x is normalized. The product Px is formed and then passed through a vector threshold function $f_s(\cdot)$. The sample-and-hold operation prevents any input change during learning. Each element of the product $w = Px$ is equal to the inner product between x and the stored exemplars (matrix rows) in the matrix P. Each element of the threshold vector function $f_s(\cdot)$ equals the maximum tolerable SNR between a pattern and the corrupting noise expressed in radians between the stored patterns and the nominal pattern. The condition for passing the threshold is equivalent to the input being within an angle at most equal to arccos(thresh) from an exemplar. The i-th element of the threshold function is equal to:

$$f_{g_i}(w_i) = \begin{cases} 0 & \text{if } 0 \leq w_i < g_i \\ w_i & \text{if } g_i \leq w_i \leq 1 \end{cases} \tag{24}$$

where g_i is the threshold for the i-th exemplar.

Hence, an input falls within a well with center some exemplar, if the threshold is exceeded for this exemplar. Notice that the above threshold function **maintains the sign**, so that two patterns with the same magnitude but opposite sign will be classified as different pattern, e.g. the network will preserve orientation by differentiating between black-and-white from white-and-black. **If preservation of the sign is not important**, w can be replaced by $|w|$ in the inequalities in the thresholding operation. After thresholding, the output vector is fed into a *maxnet* [3] which selects the maximum thresholded output, i.e. the exemplar that is closest to the input pattern. Thus, recognition is achieved. Classification is achieved by forming the inner product between the output of maxnet and the row vector $N := [\,1\ 2\ 3 \ldots N \ldots]$. If a pattern is not recognized, the outputs of maxnet are all zero and the XOR gate becomes high, thus enabling learning of a new pattern. During the learning of a new pattern, the "choose available slot" function selects the first unoccupied row of matrix P to store the new input pattern, thus creating a new well with center the new input pattern, depth d_0, and width equal to the threshold angle Θ_i ($\Theta i = \arccos(g_i)$). If one of the outputs of the maxnet is high, this indicates that the input pattern has fallen in one, or more than one, of the attraction regions of the existing wells. In this case training of the matrix P takes place by updating the center and the depth of all the wells that have nonzero threshold output. Furthermore, the stage-age (s.a.) of all wells is examined, and wells that do not meet the stage-age requirement are eliminated, thus freeing the row (*slot*) they occupied in the storage matrix.

9 Character recognition

The ability of Dignet to self-organize in the correct number of classes according to the number of different classes of patterns in the input was tested using noisy letter characters and sinusoidal signals imbedded in noise. Eight 64x64 pixel, binary characters were chosen at random. Each character was reduced into a 4x4 character using a 16x16 template, averaging the pixel values over it, and then normalizing the resulting vector. Thus, each character was represented by a 1x16 normalized vector. Noise was added to each pixel of the 16x1 vector from a zero mean, Gaussian distribution with variance determined by a prescribed SNR. The noise variance was σ^2/n, with $n = 16$ and $\sigma^2 = 10^{-[SNR/20]}$ where the SNR is in dbs. The stage age (s.a.) was taken to be three for these simulations. Simulation results with two different SNRs, 50db and 24db, are shown in figures 5 and 6.

In both cases, Dignet was able to self-organize into the correct number of patterns, eight in this case. The 3-D plots in Figures 5 and 6, demonstrate the creation of wells (classes) during the self-organization of Dignet and are recorded according to the well depth. For 50dbs very few spurious wells are generated and survived the stage age. However, the number of spurious wells increased as the SNR decreased, along with their average lifetime. For both cases, Dignet was able to classify the eight different input patterns into eight different classes (wells).

In Fig. 7 the history of the center of a well is being recorded as a function of the deviation of the center of the well from the pattern that it represents. The crosses represent the distance of (angle between) the well center associated with each input pattern from the nominal pattern, and is measured in degrees. The squares are the data points and represent the distance of an input pattern from the nominal pattern. The well width (threshold) for this particular case is set at 13^0, commensurate with the 24db SNR. Various spurious wells are created during the self-organization. However, only the center of one well gets reinforced and converges to the true pattern, its center distance from the nominal pattern approaching zero, whereas all other spurious wells get eliminated. Similar picture is obtained when different characters are presented alternately.

10 Detection of unknown number of unknown signals

An experiment was conducted using eight cosines with integer frequencies one through eight. Each cosine was sampled at the Niquist rate of the highest frequency. Thus, sample vectors of size 1x16 were generated. At each element of the vectors noise was added from a zero mean, Gaussian distribution with variance σ^2/n with $n = 16$ and $\sigma^2 = 10^{-[SNR/20]}$, determined by the specified SNR. From figure 8 it is seen that Dignet is capable of self-organization in the correct number of signals for SNR 5 and 0 dbs with a limited number of spurious classes. However, for -5 dbs, the number of spurious classes increases, their life expectancy increases, and the resolution of the correct classes

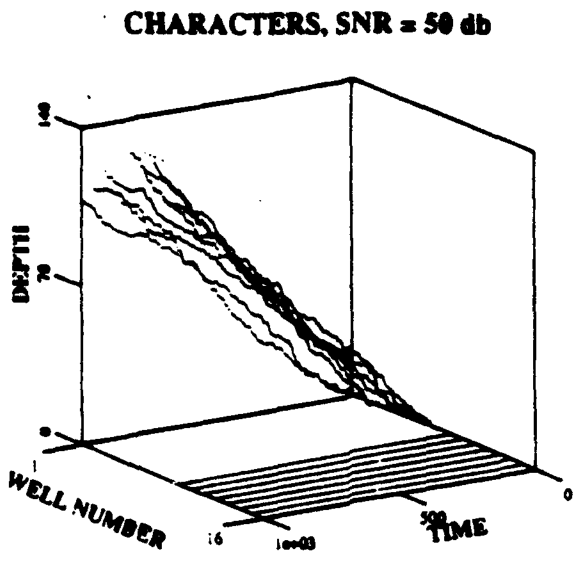


Figure 5: Space-time history of well-creation for eight different characters at 50db SNR

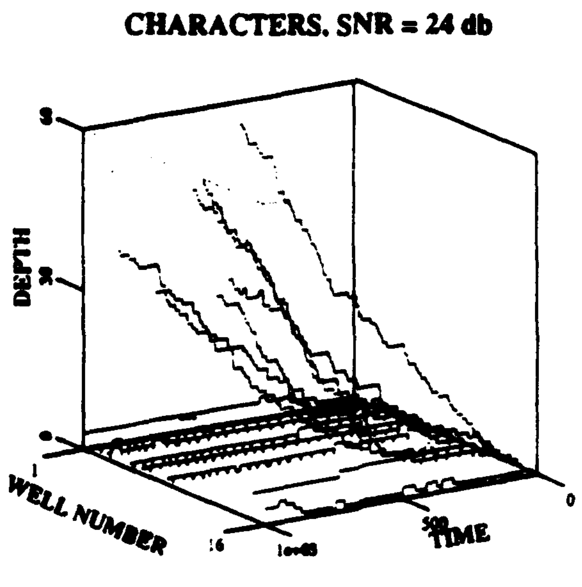


Figure 6: Space-time history of well-creation for eight different characters at 24db SNR

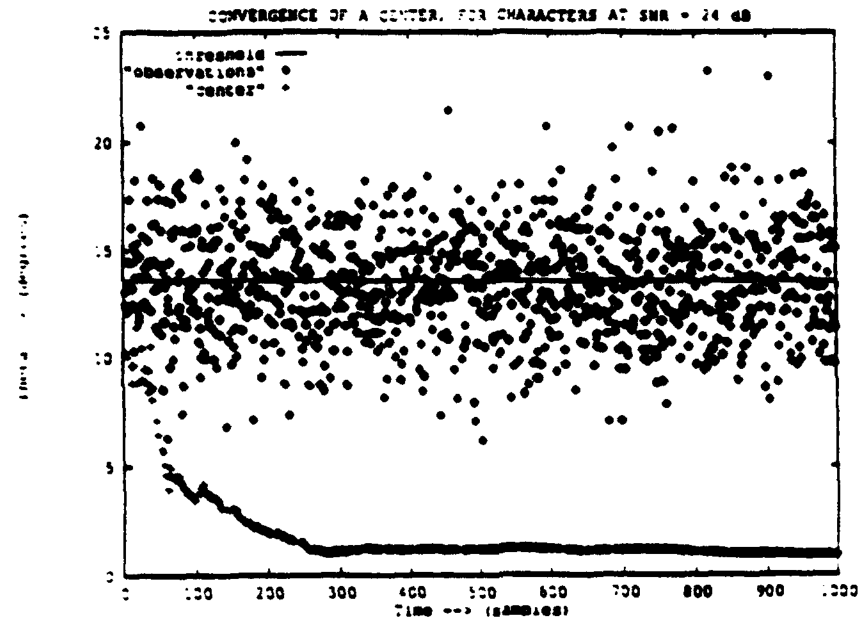


Figure 7: Convergence of a well center in Dignet for one character at 24 SNR

decreases. Fig.9. A similar picture is obtained by tracking the center of the wells for 10 db in figure 10. Obviously, only the center of one well converges to the true input cosine. Similar picture is obtained when cosines of different frequency are presented.

11 Topology of Multisensor Fusion Using DIGNET

In [12], [13], [14], [15] [16] and [17] Bayesian and Neyman-Pearson (N-P) theory for the Distributed Decision Making (DDM) problem was developed. In [18] it was shown that there exists a one-to-one topological correspondence between the Bayesian and N-P solution of the DDM problem and neural networks. Furthermore it was shown that neural networks exhibit Receiver Operating Characteristics (ROC) that are close to the optimal Likelihood Ratio Test (LRT) ROC, when trained with the proper training rule [19].

It has been shown that the DIGNET can be successfully used for detection of unknown number of unknown patterns. In this section a topology is proposed for using Dignet in Multisensor Detection.

Figure 11 shows an implementation of the parallel fusion scheme of [20], using Dignets. The signal received by each sensor is fed to a Dignet (possibly after some preprocessing), where the closest stored signal (pattern) is recognized and appears at the output "e" of the s-th sensor Dignet (fig. 3) as a vector P_s. A weighted average of the outputs (see below) is then fed to the Dignet of the fusion center, which is used as a classifier (only output "c" in fig. 3 is used).

Along with the vector outputs of the sensor Dignets, the *well depths* of the recognized patterns are fed to the weighted average stage where they are used as the weighting factors:

$$F_{in} = \sum_{s=1}^{m} d_s P_s \tag{25}$$

where F_{in} is the input vector of the fusion center, m is the number of sensors and P_s and d_s are the output vectors and depths of the sensor Dignets.

A deep well is a well that has "recognized" many patterns and a shallow well is one that most probably is spurious (created by some *outlier* signal or by pure noise). This motivates the above topology where a recognized pattern with a deeper well is taken into consideration more than another of less depth. In practice, this means that since a sensor

COSINES, SNR = 10 db

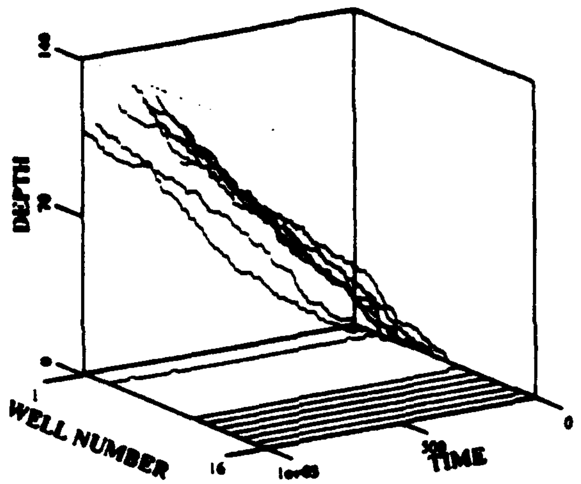


Figure 8: Space-time history of well-creation for eight different frequency cosines at 10db SNR

COSINES, SNR = -5 db

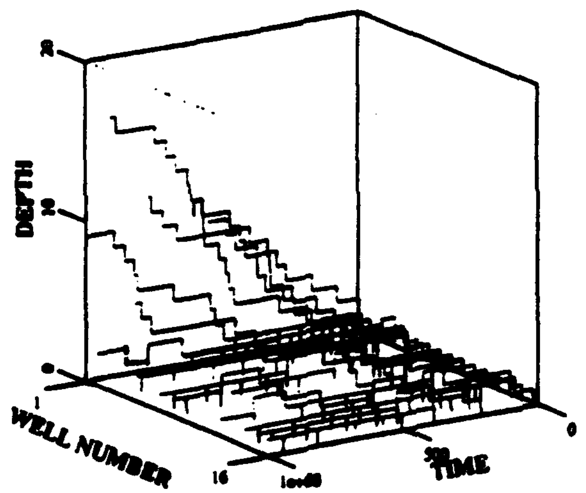


Figure 9: Space-time history of well-creation for eight different frequency cosines at -5db SNR

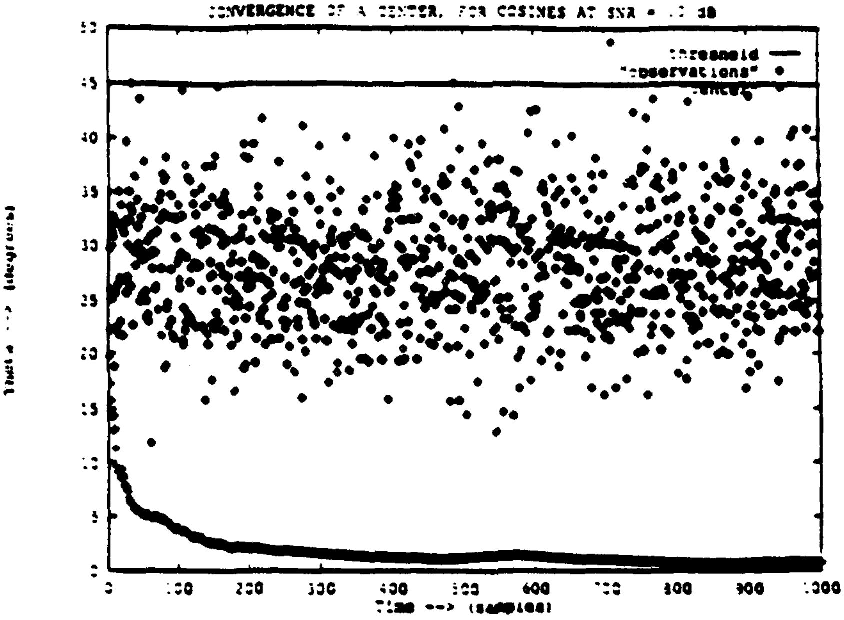


Figure 10: Convergence of a well center in DIGNET for one cosine at 10 SNR

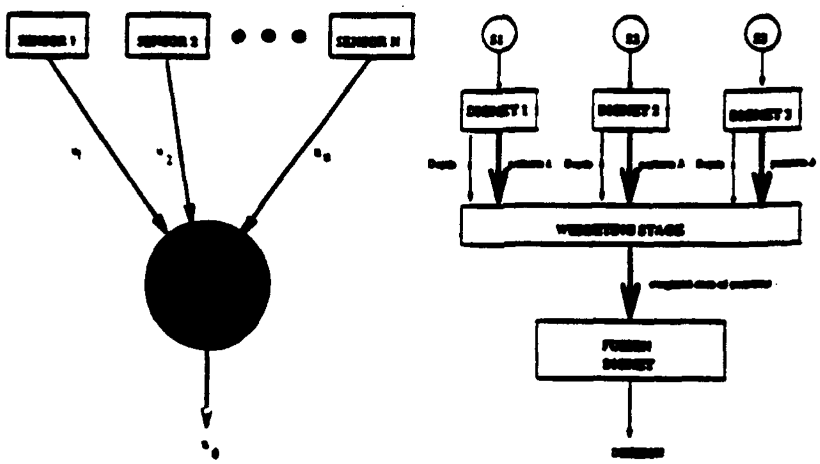


Figure 11: Parallel fusion topology and Dignet implementation

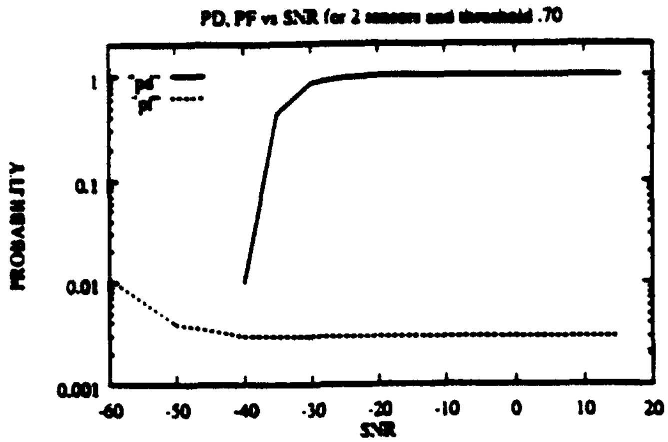


Figure 12: Probabilities of detection and false alarm as functions of SNR (2 sensors) for threshold .70

with higher SNR produces deeper wells than another with lower SNR, its output is taken into higher consideration by the fusion Dignet. thus. the Dignet fusion topology has built-in fault tolerance.

The binary case (two signals) is a special case of the case of unknown number of unknown signals. Furthermore. in the Radar detection problem there is only one signal. Hypothesis H_1 corresponds to the presense of signal plus noise and hypothesis H_0 corresponds to the presense of only noise. Without noise the absence of signal would result in a zero pattern vector which is a singularity in the director space. since it cannot be mapped on the surface of unity sphere (it cannot be normalised). In order for the Dignet to function in that case the zero vectors must be ignored (neither recognition nor training is performed).

An alternative approach is to map the zero vector by convention to some other vector. This is valid only if the signal is known so that the choice of a different director is possible for the mapping of the zero vector.

In the application of Dignet on the multisensor radar detection problem the first approach was used. i.e. the zero vector was ignored. It is thus expected that the signal will create a single "deep" well corresponding to H_1 and in the absence of signal. the noise. having random direction. is mapped on the surface of the unity sphere in such a way that no matter what is the noise distribution. the distribution on the surface is *uniform*. at least in the Gaussian noise case. This causes shallow wells to be created (uniformly) on the surface and disappear after very sort time.

For the following experiments a cosine was sampled at the Niquist sample rate and Gaussian noise was added to the sampled vector element by element. as in section 10.

In figures 12 and 13 the threshold is .70 and .85 respectively and P_F and P_D are plotted vs SNR. We notice that P_F assumes a minimum value and it cannot decrease further no matter how high the SNR is.

In figure 14 the SNR stays constant at -30 db and the P_F and P_D are plotted w.r.t. threshold. As expected they both decrease as the threshold increases and the well becomes smaller.

The case of unequal SNRs is tested in figure 15. Initially the SNR is 0 db (equal for both sensors). P_F and P_D are plotted w.r.t. time for 10^6 time slots. At time $t = .5 \times 10^6$ the first sensor breaks down and its SNR becomes -60 db.

There is no noticeable effect of the sensor malfunction in the graph. The very high noise of the broken sensor causes misclassification but the weighting stage (figure 3) causes the fusion to ignore the sensor's output. The ripple in the P_F curve is due to the small number of time samples.

In figure 16 the Receiver Operating Characteristic is given for SNR -30.

In figure 17 the case for SNR = -30 is shown for a 3 sensor fusion. The corresponding R. O. C. is shown in figure 18.

In figure 19 the case for SNR = -30 is shown for a 4 sensor fusion. The corresponding R. O. C. is shown in figure 20.

We notice that increasing the number of sensors increases the P_{D_e} for the same P_{F_e} For example for $P_{F_e} = .003$. with 2 sensors $P_{D_e} = 0.875$. with 3 sensors $P_{D_e} = 0.95$ and with 4 sensors $P_{D_e} = 0.98$.

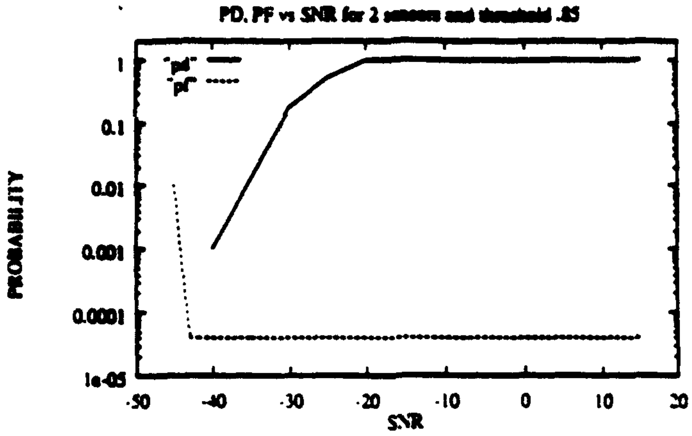


Figure 13: Probabilities of detection and false alarm as functions of SNR (2 sensors) for threshold .85

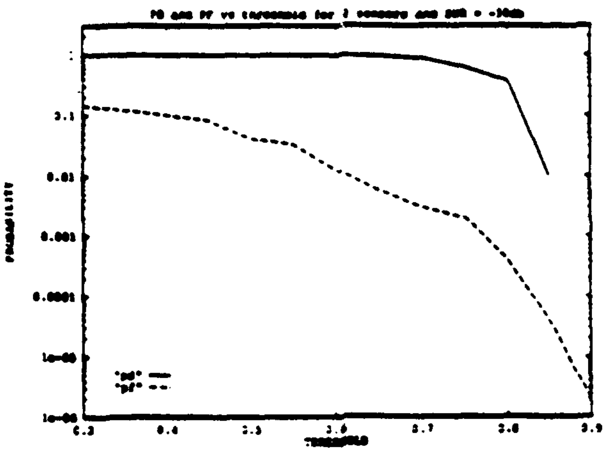


Figure 14: Probabilities of detection and false alarm as functions of threshold (2 sensors) for SNR -30 db

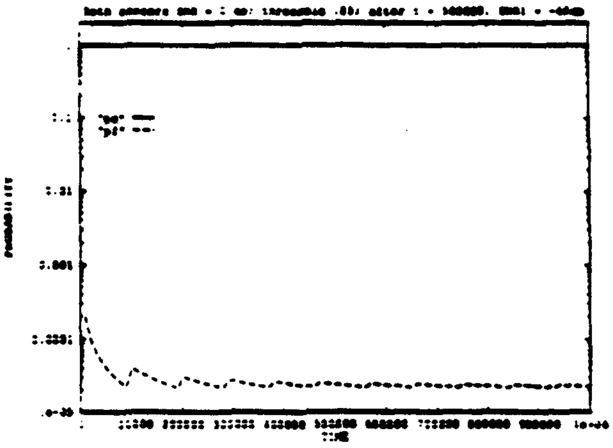


Figure 15: Probabilities of detection and false alarm as functions of time (2 sensors) for SNR 0 db. At time $t = 5 \times 10^5$ the first sensor breaks down and its SNR becomes -60db.

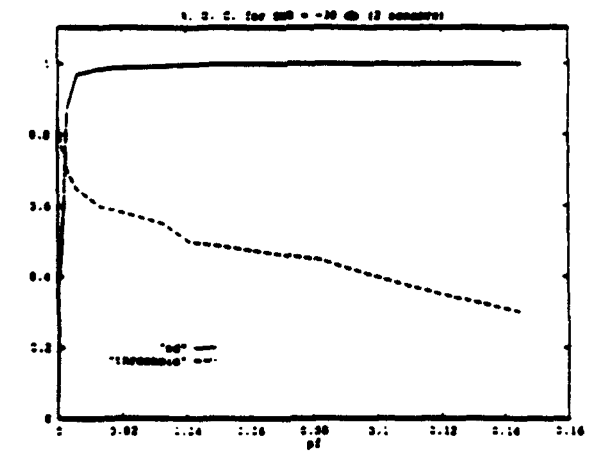


Figure 16: Receiver Operating Characteristic for Gaussian noise and SNR = -30db.

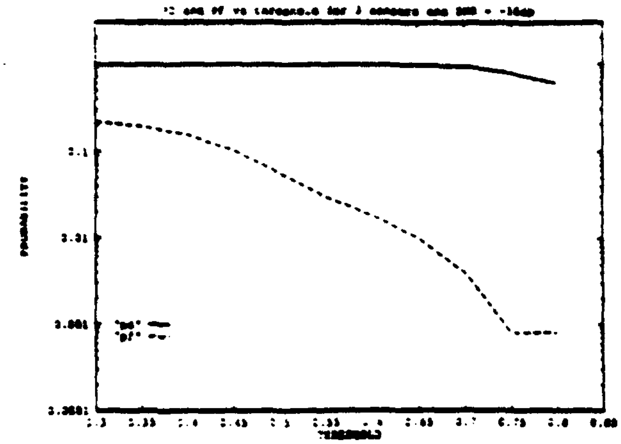


Figure 17: P_D, P_F vs threshold. Gaussian noise. SNR = -30 dB. 3 sensors

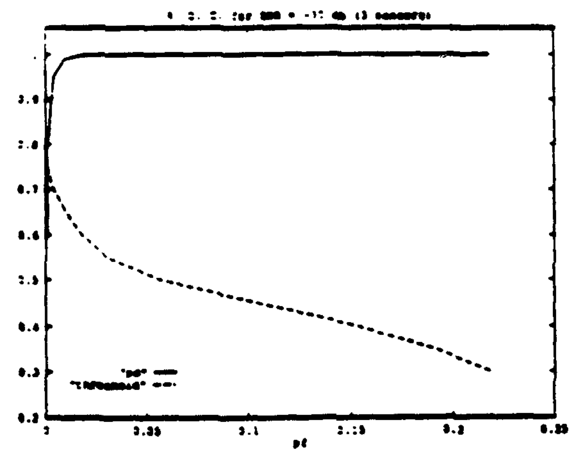


Figure 18: Receiver Operating Characteristic. Gaussian noise. SNR = -30 dB. 3 sensors

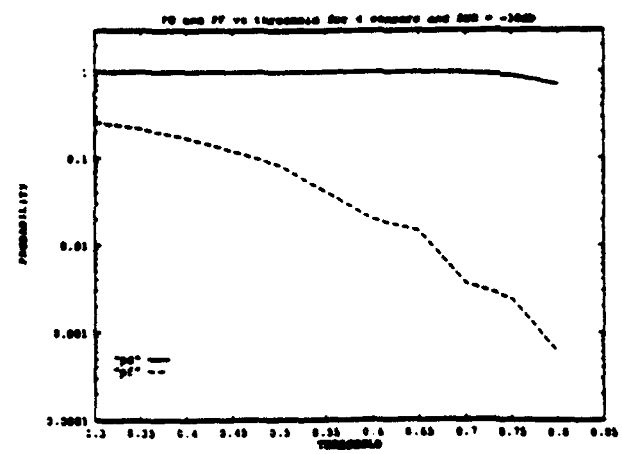


Figure 19: P_D, P_F vs threshold. Gaussian noise. SNR = -30 dB. 4 sensors

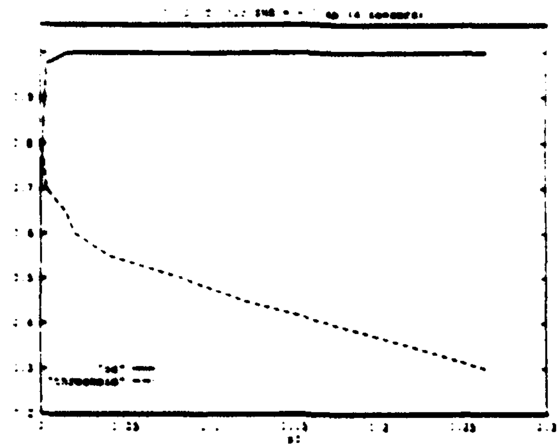


Figure 20: Receiver Operating Characteristic. Gaussian noise. SNR = -30 dB. 4 sensors

12 Conclusions

A new artificial neural network. DIGNET. was introduced for automatic pattern recognition and classification. The proposed ANN exhibits self-organization capabilities according to prescribed tolerance to noise interference. and neuron requirements that grow linearly with the size and the number of patterns that are needed to be stored. It is shown that the self-organisation algorithm of Dignet leads to stable classes that are created around patterns that are sustained in the input data over time. Dignet was tested successfully with pattern classification and signal detection paradigms.

A sensor fusion topology using DIGNETS was introduced and numerical results. for Gaussian additive noise showed that Dignet performs well under unknown statistical environments.

References

[1] John J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of National Academy of Science*. 79:2554–2558. April 1982.

[2] J. L. McClelland. D. E. Rumelhart. and the PDP Group. *Parallel Distributed Processing, Vol. 1 & 2*. The MIT Press. Cambridge. MA. 1987.

[3] Richard P. Lippmann. An introduction to computing with neural nets. *IEEE ASSP MAGAZINE*. pages 4–22. April 1987.

[4] DARPA neural network study. Oct. 1987 – Feb. 1988. Executive Summary, Lincoln Lab.. MIT. Lexington. MA. 1987.

[5] Stephen Grossberg. Nonlinear neural networks: Principles. mechanisms and architectures. In *Neural Networks. Vol. 1*, pages 17–61. Pergamon Press. 1988.

[6] Lei Zhang and S. C. A. Thomopoulos. Neural network implementation of the shortest path algorithm for traffic routing in communication networks. International Conference on Artificial Neural Networks. poster paper. June 1989.

[7] S. C. A. Thomopoulos. L. Zhang, and C. D. Wann. Neural network implementation of the shortest path algorithm for traffic routing in communication networks. In *Allerton Conference*. October 1990.

[8] S. C. A. Thomopoulos and Dimitrios K. Bougoulias. DIGNET: A self-organizing neural network for automatic pattern recognition and classification. SPIE conference on sensor fusion. Boston. MA. to appear. November 1991.

[9] Teuvo Kohonen. State of the art in neural computing. In *IEEE First International Conference on Neural Networks*. pages I-79–I-90. 1987.

[10] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. J. Wiley & Sons. New York. 1973.

[11] J. S. Balas and A. LaVigna. Convergence of a neural network classifier. In *Proceedings of 29th CDC. Honolulu. Hawaii*. pages 1735–1740. December 1990.

[12] S. C. A. Thomopoulos. R. Viswanathan. and Dimitrios K. Bougoulias. Optimal decision fusion in multiple sensor systems. In *Proceedings of the 24th Allerton Conference. Monticello. IL. Oct 1-3*. pages 984–993. 1986.

[13] S. C. A. Thomopoulos. R. Viswanathan. and Dimitrios K. Bougoulias. Optimal decision fusion in multiple sensor systems. *IEEE Transactions on Aerospace and Electronic Systems*. 23:644–653. September 1987.

[14] S. C. A. Thomopoulos. R. Viswanathan. and D. K. Bougoulias. Optimal and suboptimal distributed decision fusion. Technical Report TR-SIU-DEE-87-5. Department of Electrical Engineering, Southern Illinois University. Carbondale. IL. August 1987.

[15] S. C. A. Thomopoulos. Dimitrios K. Bougoulias. and Lei Zhang. Optimal and suboptimal distributed decision fusion. In *SPIE Technical Symposium on Optics. Electro-Optics. and Sensors. Apr. 4-8. Orlando. FL*. 1988.

[16] S. C. A. Thomopoulos. R. Viswanathan. and Dimitrios K. Bougoulias. Optimal and suboptimal destributed decision fusion. In *22nd Annual Conference on Information Sciences and Systems. Princeton University. NJ. March 16-18*. pages 886–890. 1988.

[17] S. C. A. Thomopoulos. R. Viswanathan. and Dimitrios K. Bougoulias. Optimal distributed decision fusion. *IEEE Transactions on Aerospace and Electronic Systems*. 25:761–765, September 1989.

[18] S. C. A. Thomopoulos. Decision and evidence fusion in sensor integration. In *Advances in Control and Dynamic Systems*. Academic Press. November 1991. Volumes 45. 46. 47 and 48, to appear.

[19] S. C. A. Thomopoulos. I. Pappadakis. H. Sahinoglou. and D. Bougoulias. Centralized and distributed decision making with structured adaptive networks. perceptron like and self-organizing neural networks. In *Data Fusion in Robotics and Machine Intelligence*. Academic Press. 1992. To appear.

[20] D. K. Bougoulias. *Distributed Decision Making with Bayesian and Neural Network Approaches*. PhD thesis. University of Southern Illinois. Carbondale. IL. June 1991.

Volume 3 of 3
ISBN 0-7803-0227-3

I J C N N

S I N G A P O R E

'91

INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS

WESTIN STAMFORD AND

WESTIN PLAZA

18-21 NOVEMBER 1991 SINGAPORE

INNS

NEURAL NETWORK IMPLEMENTATION OF THE SHORTEST PATH ALGORITHM FOR TRAFFIC ROUTING IN COMMUNICATION NETWORKS

Stelios C.A. Thomopoulos[1,#], Lei Zhang[●], Chin Der Wann[#]

[#] Decision and Control Systems Laboratory
Department of Electrical & Computer Engineering
The Pennsylvania State University, University Park, PA 16802

[●] Department of Electrical Engineering
The University of Maryland, College Park, MD 20740

Abstract

A neural network computation algorithm is introduced to solve for the optimal traffic routing in a general N-node communication network. The algorithm chooses multilink paths for node-to-node traffic which minimize a certain cost function (e.g. expected delay). Unlike the algorithm introduced earlier in this area, knowledge of the number of links (hops) between each origin-destination pair is not required by the algorithm, therefore it can be applied to a more variable length path routing problems. The neural network structure for implementing the algorithm is a modification to the one used by the Traveling Salesman algorithm. Computer simulations in a nine- and sixteen-node grid network show that the algorithm performs extremely well in single and multiple paths.

I. Introduction

The computational power and the speed of collective analog networks of neurons in solving optimization problems have been demonstrated by Hopfield and Tank [1]-[3] through the famous "Traveling Salesman Problem". A similar procedure can be applied to solve a number of optimization problems [6]. In order to solve a practical optimization problem using a neural network structure, it is necessary to find algorithms for determining the connections and weights of the neural network so that it converges to the appropriate answer. In this paper, we suggest a neural network structure that can determine the optimal route for node-to-node traffic in an N-node communication network. The structure is an implementation of the so called "Shortest Path Routing Algorithm" in which a route is selected for every origin-destination (OD) pair such that the transmission cost is minimized if data is transmitted along this route.

The main function performed by a routing algorithm is the selection of routes for various origin-destination pairs. There are two main performance measures that are substantially affected by the routing algorithm, the throughput (quantity of service) and the average delay (quality of service). A good routing algorithm should select the routes which have minimum average delay (thus allow more traffic into the network). In the shortest path algorithm, a cost is associated with every link in the network. In most cases, the cost is proportional to the delays. The objective is to find a multilink path joining two nodes that has minimum total cost. Different implementations of the shortest paths algorithm, in both synchronous and asynchronous fashion, are available [4]. In this paper we consider two different NN implementations of the shortest paths algorithm using the actual delay and the derivative delay as cost functions. The neural network structure of the algorithm was first introduced by Rauch and Winarske [5]. Their method, however, has serious limitations. It can find the shortest path for a given OD pair only when the number of links that the path contains is known, which is an unrealistic assumption. A modified structure is suggested in the present paper so that the algorithm can work for arbitrary and unknown number of links in a given OD pair. The NN that is presented in this paper was first introduced in [7].

II. Problem Statement

Consider a N-node network and assume that the connectivity of the network is known. Let c_{ij} denote the capacity of the link connecting node i with node j. If there is no direct connection between node i and j, $c_{ij} = 0$. Therefore, the network can be described by an NxN capacity matrix C with entries c_{ij}. In addition, if every link in the network is a two-way link and has the same capacity in each direction, C is symmetric.

Our problem is to find the path connecting origin and destination nodes that minimizes a cost function such as the expected delay. Since the expected delay across a link is a function of the link capacity c_{ij} and the actual link traffic f_{ij}, several functions can be used to calculate the link cost [4], [5]. For example, the link cost w_{ij} can be determined by

$$w_{ij} = f_0 + [\Sigma f_{ij} / (c_{ij} - \Sigma f_{ij})]^p \qquad (1.1)$$

where f_0 is the transmission time for each link, and Σf_{ij} is the total flow from all OD-pairs on the link ij. The exponent p can take any positive value, but commonly used values are 1 or 2. The value p = 1 was used in the simulations. The link capacity $c_{ij} - \Sigma f_{ij}$ in (1.1) is the residual capacity in the network when paths for multiple OD-pairs are considered. When the optimal paths for multiple OD-pairs are determined sequentially, the residual link capacity is determined by

$$c_{ij} - \Sigma f_{ij} = c_{ij} - \Sigma f_{ij}(\text{previous OD pairs}) - \Sigma f_{ij}(\text{current OD-pair}) \qquad (1.2)$$

With p = 1 in (1.1), two different approaches were taken to solve for the shortest path. In the first approach, which will be referred to as *delay cost* approach, the link cost was computed directly using (1.1). In the second approach, which will be referred to as *derivative delay cost* approach, the link cost was equated to the derivative of w_{ij} in (1.1), i.e.

[1] *This work was partially supported by SDIO/IST under Contract N-00014-k-0515 managed by ONR*

$$w_{ij} = f_0 + c_{ij} / (c_{ij} - \Sigma f_{ij})^2 \qquad (1.3)$$

which is the link cost that is used in the conventional, optimal solution of the shortest path problem assuming convex and double-differentiable delay (cost) function [4]. The differences in the numerical solutions obtained under the two cost functions (1.1) and (1.3) are discussed in the simulations.

Let the NxN matrix W with entries w_{ij} denote the cost matrix associated with the network. Notice that if there is no direct link between node i and j, $w_{ij} = \infty$ ($c_{ij} = 0$). If this is the case, a very large number is assigned to w_{ij} in the simulation.

To illustrate the problem, consider the 5-node network in Figure 1. The number beside each link represents the corresponding link cost (w_{ij}). The cost matrix W associated with this network is given in Table 1, where L is some large positive number.

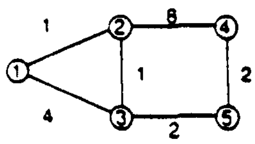


Figure 1 A 5-Node Network

Table 1 Cost Matrix for the 5-Node Network of Fig. 1

	1	2	3	4	5
1	L	1	4	L	L
2	1	L	1	8	L
3	4	1	L	L	2
4	L	8	L	L	2
5	L	L	2	2	L

The shortest path from node 1 to node 5 is obviously 1-2-3-5 and the minimum total cost is $w_{12} + w_{23} + w_{35} = 1+1+2 = 4$.

In the next section we reformulate the shortest path algorithm using a neural network structure.

III. Neural Network Computation Algorithm

In their paper [5], Rauch and Winarske suggested that the solution of the shortest path algorithm can be represented by a 2-dimensional neuron array $V = (V_{ij})$ with each output of the neuron in the array having value $V_{ij} = 0$ or 1. The number of rows in the array is equal to N, the number of nodes in the network, and the number of columns is equal to the number of nodes that the path contains. For the 5-node network of Figure 1, the shortest path connecting node 1 and node 5 can thus be represented by

	1	2	3	4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	0
5	0	0	0	1

(2)

It is obvious that for the array to represent a valid path, there can be only one nonzero entry in each column and there can be at most one nonzero entry in each row (this condition is different from the one required by the TSP problem). An nonzero entry in the ijth position of the array can be interpreted as "node i is the jth node in path". Using this representation, a total NxM neurons are needed to represent all the paths having length (number of nodes in the path) M. Given an OD pair, the first and the last column of the array are fixed, so there are Nx(M-2) active neurons in the array which are free to be updated.

As we have mentioned in the previous section, this representation has its limitations. The problem with this representation is that if we do not know how many nodes the shortest path would contain, i.e. M is unknown. Rauch and Winarske assumed that the minimum number of links between a given OD pair could be obtained in advance from the capacity matrix C, in which case M is equal to that number plus one. However, by choosing M this way, we may not be able to find the shortest path because it is possible that a longer path can have lower total cost than that of a shorter path. In our 5-node example, the minimum number of links between node 1 and 5 is 2. If we choose M = 3, the 5x3 array can only give the path that contains 3 nodes, which is 1-3-5 with cost 6. We know though from the previous discussion that the shortest path is 1-2-3-5 with cost 4. It is obvious that the solution given by Rauch-Winarske's (R-W) method is not the correct one.

To overcome the limitations of the R-W method, we fix the number of columns in the array at N, which is the maximum possible number of nodes any path could contain in an N-node network. By doing so, the neuron array (NxN now) can represent all the paths containing N nodes. Since most of the paths have length less than N, we should convert these paths into length N paths and maintain their total cost at the same time in order for them to be represented by the NxN array. This can be achieved *by adding some zero-cost pseudo links to those "shorter" paths, i.e. paths with less-than-N-links, until their length is equal to N. To implement this idea, for each node we introduce one zero-cost pseudo link that connects the node to itself.* The traffic can then circle at any node along the path through these pseudo links without increasing the total cost of the path. For the 5-node example of Fig. 1, the network after introducing 5 pseudo links is shown in Fig. 2. Table 2 gives the associated cost matrix.

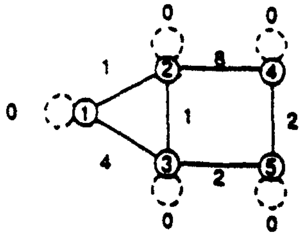


Figure 2 A 5-Node Network with Pseudo Links

Table 2 Cost Matrix for the 5-Node Network of Fig. 2

i	1	2	3	4	5
1	0	1	4	L	L
2	1	0	1	8	L
3	4	1	0	L	2
4	L	8	L	0	2
5	L	L	2	2	0

By comparing Table 1 with Table 2, one can see that the only difference between the two cost matrices is that the diagonal elements now become zero instead of a large number L. Using this modified representation, one of the possible solutions to the 5-node network problem with node 1 being the origin and node 5 the destination is

i	1	2	3	4	5
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	0
4	0	0	0	0	0
5	0	0	0	0	1

(3)

(3) shows that the shortest path between node 1 and 5 is 1-1-2-3-5, which can be interpreted as 1-2-3-5. Note that the representation of the shortest path is not unique; solutions 1-2-2-3-5, 1-2-3-3-5, and 1-2-3-5-5 all represent the same path.

For a solution to be valid, we require that there is only one nonzero entry in each column and the total number of nonzero entries in the array is equal to N. Under these constraints, the energy function associated with the network can be defined as

$$E = (A/2)\sum_k\sum_i\sum_j V_{ik} w_{ij} V_{j,k+1} + (B/2)\sum_k\sum_i\sum_j V_{ik} V_{jk} + (C/2)\Big(\sum_i\sum_j V_{ij} - N\Big)^2 \tag{4}$$

where the first triple summation gives the total cost from the origin to destination; the second and third terms are the constraints imposed on the output on the neuron array to make it converge to a valid path; A,B, and C are positive enforcement factors.

From Equation (4) we can obtain the connection weight between the ij^{th} neuron and the mn^{th} neuron in the array

$$T_{ij,mn} = -A\, w_{im}(\delta_{n,j+1} + \delta_{n,j-1}) - B\,\delta_{jn}(1-\delta_{im}) - C \tag{5}$$

where δ_{ij} is the Kronecker's delta, i.e. $\delta_{ij} = 1$ if $i=j$ and $\delta_{ij} = 0$ if $i \neq j$.

The state of the ij^{th} neuron, y_{ij}, can be described by the differential equation

$$dy_{ij}/dt = -y_{ij}/\tau + \sum_m\sum_n T_{ij,mn} V_{mn} + I_{ij} \tag{6}$$

and

$$V_{ij} = g(y_{ij}) = [1 + \tanh(y_{ij}/y_0)]/2 \tag{7}$$

$$I_{ij} = Cn \quad \text{(input bias term)} \tag{8}$$

for $1 \leq i \leq N, 2 \leq j \leq N-1$.

In high gain limit, the output of the neuron, V_{ij}, is close to 0 or 1, and the energy function defined by (4) will be minimized (it could be a local minimum) when the system reaches its steady state.

IV. Simulation Results

The neural network routing algorithm developed in the previous section was simulated using 9-node and 16-node networks with different link cost assignments. The 9-node grid network shown in Figure 3 was used for the first pilot simulation. All links were assumed to be two-way links and have the same capacity. Under this assumption, the capacity matrix C is symmetric. We also assumed that the initial link cost w_{ij} is inverse proportional to the link capacity c_{ij} because we do not have any knowledge about the link flow (traffic) f_{ij} when we first start the algorithm. So, the cost matrix W is also symmetric and all links have the same cost. The diagonal elements of W, which correspond to pseudo links, are all zero, and a large number is assigned to the elements which correspond to "open" links. The cost matrix used in the pilot simulations is given in Table 3.

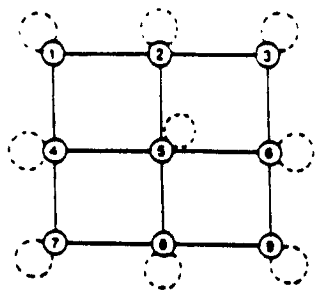


Figure 3 A 9-Node Grid Network with Pseudo Links (dashed lines)

Table 3 Cost Matrix for the 9-Node Grid Network of Fig. 3

	1	2	3	4	5	6	7	8	9
1	0	2	20	2	20	20	20	20	20
2	2	0	2	20	2	20	20	20	20
3	20	2	0	20	20	2	20	20	20
4	2	20	20	0	20	20	2	20	20
5	20	2	20	2	0	2	20	2	20
6	20	20	2	20	2	0	20	20	2
7	20	20	20	2	20	20	0	2	20
8	20	20	20	20	2	20	2	0	2
9	20	20	20	20	20	2	20	2	0

In the pilot simulations only one OD-pair was considered for the given cost matrix. For each given OD pair, the first and the last column in the neuron array are fixed. The states of the rest N(N-2) (= 63 in the 9-node network) active neurons are updated according to the steady state expression of Equations (6) and (7). The initial value of the output of each active neuron is a random number uniformly distributed in [0, 2/N] such that

$$E\{\sum_i \sum_j V_{ij}\} = N \tag{9}$$

We start the network with low gain, i.e., the slope of the hyperbolic tangent curve in Equation (7) is small (y_0 large). This choice would allow the system to find better minima of the energy surface. After 100 iterations, we start slowly increasing the gain (decreasing y_0) until the system converges and the values of V_{ij} are near 0 or 1. The results for the 9-node pilot study were obtained with the following parameters.

$A = 20$, $B = C = 500$, $n = 9.5$, $\tau = 1$

$y_0 = 250$ (initial), $y_0 = 20$ (final)

The algorithm is sensitive to these parameters, since a bad operating point may result in divergence (oscillation).

Table 4 shows the shortest path found by the algorithm between node 1 and node 9; (a) is the initial condition, (b) is the result after 100 iterations and (c) the result after 200 iterations (final result). The shortest path found is 1-1-2-2-2-2-5-6-9, which can be interpreted as 1-2-5-6-9. Table 5 gives similar results for a different OD pair.

Because of the symmetry of the grid network, the shortest path is not unique for some OD pairs, which makes the convergence more difficult. The algorithm will find one of the shortest paths depending on the initial conditions. In the simulation, we also noticed that if the gain is fixed at a higher value right from the beginning, the system is very easy to get stuck at some local minima. By starting at low gain and slowly increasing it, we have, so far, been able to reach the global minimum on all single-OD-pair trials for the pilot 9-node grid network.

In practice, the link cost w_{ij} in a communication network depends on the actual traffic going through that link. To obtain the actual traffic distribution for the entire network, the algorithm should be repeated for every OD pairs (there are Nx(N-1) of them). After the actual traffic conditions in the network become available, the cost matrix W can be updated by using equations (1.1) and (1.2). The algorithm is then repeated for each OD pair again, and the optimal path is found for each OD pair that will prevent some links from becoming too crowded. By so repeating the algorithm, the ANN could eventually obtain the optimal flow distribution for the network in the sense that the expected delay on the entire network is minimized for a given set of link capacities. This approach was used to obtain shortest paths for multiple OD pairs in a 9- and 16-node networks.

After the pilot simulation was successfully completed, the ANN algorithm was tested in multiple OD-pairs in both 9-node and 16-node networks. In the 9-node network, the algorithm was tested with four different OD-pairs. Each link was assumed to have normalized capacity 0.5, whereas the flow on each OD-pair was taken to be 0.1. The "optimal" paths were obtained sequentially by presenting to the ANN one OD-pair at a time. The initial conditions on A, B, C, n, and t that were used in the pilot simulation, were also used in these simulations. The annealing temperature schedule was slightly different: the initial temperature was kept the same at y_0 (initial) = 250, but the final temperature was set at y_0(final) = 30.70193 for all trials. After the network converged to an "optimal" path for a given OD-pair, the link cost was updated according to Eq. (1.1) with p = 1. The four chosen OD-pairs were: A = (1,8), B = (2,8), C = (4,2), and D = (7,3) [The first number in the parentheses indicates the origin, while the second the destination]. For each OD-pair, convergence was achieved after 200 iterations, in agreement with the pilot simulation. The initial and final neural activations for each OD-pair are shown in Table 6. In this particular simulation, the order that the OD-pairs were presented in the ANN was ABCD and a *single path* was assumed to carry all the traffic for each OD pair. The same initial conditions (neural activation) were used for each OD-pair. The "optimal" path that was obtained was A = (1-4-5-8), B = (2-5-8), C = (4-1-2), and D = (7-4-5-2-3), with total cost 16.63968. This path is not the overall optimal path which has cost 15.42510 (see Table 7), but is very close to it.

In order to determine the effect of the sequence at which the different OD-pairs are presented to the neural network, all possible permutations in the sequence of the four OD-pairs were presented and the "optimal" paths were recorded. Table 7 summarizes the different "optimal" paths and the frequency they occurred. When the same initial conditions were used for each OD-pair, the set of paths 1, with total cost 16.63968, very close to the optimal set of paths 10 with cost 15.42510, was obtained 66.67% of the times. When the initial conditions (neural activations) for each OD-pair were chosen randomly, the frequency of path set 1 dropped to 29.17%. However, the frequency of the optimal path set 10 increased from 0.0%, that it was when the same initial conditions were used, to 12.50%. The effect of the initial conditions is currently investigated. From the results obtained so far, it appears that the different initial conditions result in a more even distribution of the path sets among low cost path sets than the distribution of the path sets obtained with fixed initial conditions. Table 8 summarizes the correspondence between the sequence with which the four OD-pairs were presented to the network and the path set that the NN converged to under fixed initial conditions and different initial conditions. The numbers of the path sets correspond to the path set numbers of Table 7.

In order to determine the stability of the "optimal" path sets, two OD-pairs were alternatingly presented to the NN and the path sets were recorded. The chosen OD-pairs were A = (1,9) and B = (8,3). The link capacity was kept the same, i.e. 0.5 units per link, but the input data flow was raised to 0.25 data units. Starting with zero initial neural activation, fixed for each OD-pair, the NN converged to a stable solution in one iteration. Furthermore, it converged to the same path set, irrespective of which OD-pair was presented first (columns 1 and 2, Table 9). When the initial neural activation were random but fixed for all OD-pairs, the solution was stabilized in a few iterations, columns 3 and 4 in Table 9. The same path sets were obtained irrespective of what OD-pair was presented first. However, when different random initial activation was used each time a new OD-pair was presented, the path sets stabilized after a few presentations at slightly different set paths, depending on which OD-pair was presented first. In this particular experiment, all the different path set that were obtained are equivalent from cost point of view.

To test the ability of the ANN to optimize the network performance further by creating multi-path routes for multiple OD pairs, a comparative study was conducted by allocating different percentages on the total flow on each path and repeating the algorithm by interleaving the different OD pairs until the total traffic from all OD-pairs was accommodated. The simulations were conducted using both the *delay link cost* [Eq. (1.1)] as well as the *derivative delay link cost* [Eq. (1.3)]. For a single OD pair but different percentage of traffic allocation at each "shortest" path, the results for the two different cost functions applied to a 9-node network are summarized in Table 10 and Fig. 4. From these results, it can be seen that smaller increments per iteration result is lower total cost, in general. Furthermore, the derivative delay cost function (1.3) yields paths that slightly outperform those obtained by the delay cost function (1.1) for most increments. However, the differences are not so significant. One advantage of the derivative delay cost function is that the number of loops observed is the "shortest" paths was eliminated completely in the run cases. A small percentage of "shortest" paths, usually less than 5%, occasionally contained loops when the delay cost function was used instead. The existence of loops is currently under investigation.

An identical simulation to the one described in the previous paragraph was conducted for three OD pairs in a nine node network. The results for the delay and derivative delay cost functions are summarized in Fig.s 5 and 6 respectively. Similar conclusions to the single-path experiment can be drawn; lower increments per iteration result in lower total cost. The derivative delay cost yields slightly better results than the delay cost function itself. Analytical statistics of the number of times each path appeared as different increments of flow were used to obtain the shortest paths are given in Tables 11 and 12 for 100%, 25%, and 1% increments per iteration. The amount of flow each path carries is also shown on the tables. As it is seen, most of the traffic flow is concentrated in a few "good" paths as the size of flow increment decreases. Furthermore, the derivative delay cost yields a slightly lower final cost (delay) than the delay cost.

The NN routing algorithm was also tested in a 16-node square grid network with link capacity 0.5 units, the same as in the 9-node network. Four OD-pairs were used to test the NN. The test OD-pairs were: A = (1,8), B = (2,12), C = (14,4), D = (1,13). When the OD-pairs were presented to the NN in the ABCD sequence, the optimal path set was obtained after 200 iterations for each OD-pair, Table 13. The same annealing schedule as in the 9-node case was used. Note that the path set in Table 13 is globally optimal. Due to space limitations, initial conditions, intermediate results after 100 iterations, and final results after 200 iterations are only given for OD-pair 4. For the other three OD-pairs only cumulative, final results are given. The sensitivity of the solution to the order at which the different OD-pairs are presented in the NN is being investigated. Furthermore, the appearance of paths that contain closed loops which seem to appear when the cost of looping is low and the path of the initially presented OD pair splits the network graph into two separate subgraphs, is also being investigated. Additional details on the simulation results on a 16-node network can be found in [8].

V. Conclusions

In this paper, a neural-based computational algorithm has been developed for solving optimal traffic routing problems in communication networks. The key idea in this algorithm is the introduction of pseudo links which allow the extension of any path to a length N path so that it can be represented by an NxN neuron array. The proposed NN algorithm can be used to obtain optimal routes for multiple OD-pairs by presenting to the NN one OD-pair at a time. The sequential presentation of OD-pairs in the NN guarantees a unique path for each OD-pair. Once a "shortest" path is obtained it can be used to carry the entire traffic for a given OD pair, provided its capacity is not exceeded. A more even distribution of the flow from different OD pairs using multi-paths is obtained by allocating only a percentage of the total flow from a given OD pair to a "shortest" path, and repeating the algorithm by interleaving the different OD pairs. An implementation of the algorithm using incremental, circular presentations that can allow multiple paths per OD-pair has been tested numerically and found to reducing the routing cost (i.e. total delay). Computer simulation results on a 9-and 16- node grid networks show that the algorithm performs well when the slope of the nonlinearity curve (characteristics of the neuron) is slowly increased during iterations. The "optimal" path sets were found to be close to the global optimal and be stable independent of the sequence the OD-pairs were presented to the NN. Simulation results in a 16-node network indicate that the NN algorithm continues to perform well in larger networks. The performance of the algorithm in upscaled, multi-node networks with different connectivity is presently being evaluated.

References

[1] J.J. Hopfield and D.W. Tank, *"Neural" Computation of Decisions in Optimization Problems*, Biological Cybernetics, Vol. 54, pp. 141-152, 1985.

[2] J.J. Hopfield, *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*, Proc. Natl. Acad. Sci. USA, Vol. 79, pp. 2554-58, April 1982.

[3] J.J. Hopfield, *Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons*, Proc. Natl. Acad. Sci. USA, Vol. 81, pp. 3088-3092, May 1984.

[4] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, 1987.

[5] H.E. Rauch and T. Winarske, *Neural Networks for Routing Communication Traffic*, IEEE Control Systems Magazine, pp. 26-30, April 1988.

[6] E.W. Page and G.A. Tagliarini, *Algorithm Development for Neural Networks*, SPIE Symposium on Innovative Science and Technology, Los Angeles, CA, Jan. 1988.

[7] L. Zhang, and S. C. A. Thomopoulos, *Neural Network Implementation of the Shortest Path Algorithm for Traffic Routing in Communication Networks*, Joint International Conference on Neural Networks, *IJCNN '89, June 1989, Washington, D.C.*, Poster Paper Proceedings.

[8] S. C. A. Thomopoulos, and C. D. Wann, *"Neural Network Implementation of the Shortest Path Algorithm for Traffic Routing in Communication Networks*, Technical Report, April 1990, Decision and Control Systems Laboratory, Dept. of ECE, The Pennsylvania State Univ., University Park, PA 16802.

Table 4 Simulation Results for the 9-Node Grid Network of Fig. 3 (Origin = Node 1; Destination = Node 9)

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9
1	1.000	0.060	0.222	0.086	0.011	0.114	0.088	0.208	0.000
2	0.000	0.127	0.080	0.211	0.079	0.138	0.082	0.134	0.000
3	0.000	0.010	0.089	0.215	0.001	0.077	0.196	0.191	0.000
4	0.000	0.066	0.197	0.049	0.089	0.020	0.067	0.113	0.000
5	0.000	0.164	0.011	0.171	0.057	0.068	0.115	0.015	0.000
6	0.000	0.028	0.071	0.085	0.216	0.060	0.068	0.159	0.000
7	0.000	0.144	0.127	0.176	0.113	0.100	0.054	0.135	0.000
8	0.000	0.188	0.067	0.019	0.106	0.081	0.081	0.207	0.000
9	0.000	0.008	0.066	0.158	0.080	0.029	0.015	0.209	1.000

(a) Initial Conditions

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9
1	1.000	0.661	0.160	0.086	0.067	0.008	0.008	0.001	0.000
2	0.000	0.289	0.752	0.790	0.767	0.733	0.016	0.008	0.000
3	0.000	0.008	0.011	0.064	0.049	0.009	0.088	0.001	0.000
4	0.000	0.012	0.004	0.001	0.001	0.009	0.001	0.006	0.000
5	0.000	0.008	0.012	0.054	0.095	0.149	0.756	0.009	0.000
6	0.000	0.000	0.001	0.001	0.001	0.011	0.012	0.634	0.000
7	0.000	0.000	0.000	0.000	0.000	0.001	0.002	0.001	0.000
8	0.000	0.000	0.000	0.001	0.001	0.008	0.003	0.303	0.000
9	0.000	0.000	0.000	0.000	0.000	0.001	0.011	0.019	1.000

(b) After 100 Iterations

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9
1	1.000	0.994	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.000	0.000	0.994	0.999	0.999	0.995	0.000	0.000	0.000
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.940	0.000	0.000
6	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.937	0.000
7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000

(c) After 200 Iterations

Shortest Path: 1-1-2-2-2-2-5-6-9 (1-2-5-6-9)

Table 5 Simulation Results for the 9-Node Grid Network of Fig.3 (Origin = Node 3; Destination = Node 8)

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9
1	0.000	0.060	0.222	0.086	0.011	0.114	0.088	0.208	0.000
2	0.000	0.127	0.080	0.211	0.079	0.138	0.082	0.134	0.000
3	1.000	0.010	0.089	0.215	0.001	0.077	0.196	0.191	0.000
4	0.000	0.066	0.197	0.049	0.089	0.020	0.067	0.113	0.000
5	0.000	0.164	0.011	0.171	0.057	0.068	0.115	0.015	0.000
6	0.000	0.028	0.071	0.085	0.216	0.060	0.068	0.159	0.000
7	0.000	0.144	0.127	0.176	0.113	0.100	0.054	0.135	0.000
8	0.000	0.188	0.067	0.019	0.106	0.081	0.081	0.207	0.000
9	0.000	0.008	0.066	0.158	0.080	0.029	0.015	0.209	0.000

(a) Initial Conditions

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9
1	0.000	0.006	0.036	0.049	0.086	0.008	0.004	0.001	0.000
2	0.000	0.680	0.771	0.780	0.761	0.625	0.100	0.006	0.000
3	1.000	0.264	0.108	0.066	0.009	0.008	0.004	0.001	0.000
4	0.000	0.000	0.001	0.001	0.001	0.008	0.011	0.006	0.000
5	0.000	0.006	0.009	0.067	0.109	0.353	0.757	0.703	0.000
6	0.000	0.011	0.001	0.001	0.001	0.008	0.011	0.004	0.000
7	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.009	0.000
8	0.000	0.000	0.001	0.001	0.001	0.008	0.220	0.187	0.000
9	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.009	0.000

(b) After 100 Iterations

Node	V1	V2	V3	V4	V5	V6	V7	V8	V9
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.000	0.979	0.998	0.998	0.998	0.978	0.000	0.000	0.000
3	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.977	0.975	0.000
6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

(c) After 200 Iterations

Shortest Path: 3-2-2-2-2-2-5-5-8 (3-2-5-8)

Table 6 Simulation results from a 9-Node Grid Network of Fig. 3
with four OD-pairs (1,8), (2,8), (4,2), (7,3)
Cummulative Results

OD-pair in order of presentation	*Initial Temperature*	*Temperature after 100 iterations*	*Final Energy after 200 iterations*	*Selected Path at the end of annealing*
(1,8)	250	30.70193	12.94737	1-4-5-8
(2,8)	250	30.70193	13.87045	2-5-8
(4,2)	250	30.70193	14.79352	4-1-2
(7,3)	250	30.70193	16.63968	7-4-5-3

The total energy is: 16.63968

Initial and final (200iterations) neural activation for the third OD-pair (4,2)

Initial neural activation

0.000	0.120	[illegible]	0.014	0.196	[illegible]	0.078	0.086	0.020
0.000	[illegible]	0.116	[illegible]	0.011	[illegible]	0.176	0.047	1.000
0.000	[illegible]	0.219	0.216	0.077	0.047	0.087	0.141	0.000
1.000	0.107	[illegible]	[illegible]	0.170	0.179	[illegible]	0.140	0.000
0.000	[illegible]	[illegible]	0.142	0.116	0.145	0.065	0.120	0.000
0.000	[illegible]	0.120	0.150	0.160	0.060	0.117	[illegible]	0.000
0.000	0.154	0.125	0.011	[illegible]	0.170	[illegible]	[illegible]	0.000
0.000	0.164	0.145	0.140	0.180	[illegible]	0.180	0.191	0.000
0.000	0.158	0.171	0.097	0.164	0.212	0.168	0.011	0.000

Neural activation after 200 iterations

0.000	0.000	0.000	0.000	0.000	[illegible]	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	[illegible]	[illegible]	1.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.000	[illegible]	[illegible]	[illegible]	[illegible]	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Results	Selected OD pairs				Path cost	% (with same initial condition)	% (with different initial conditions)
	A=(1,8)	B=(2,8)	C=(4,2)	D=(7,3)			
1	1-4-5-8	2-5-8	4-1-2	7-4-5-2-3	16.63968	66.67%	29.17%
2	1-4-5-8	2-5-8	4-5-2	7-4-5-2-3	19.01330	12.50%	---
3	1-4-5-8	2-5-8	4-5-2	7-4-5-6-3	17.68074	8.33%	---
4	1-2-5-8	2-1-4-5-8	4-5-2	7-4-5-2-3	19.93636	8.33%	---
5	1-2-5-8	2-1-4-5-8	4-5-2	7-4-5-6-3	18.60382	4.17%	---
6	1-4-5-8	2-5-8	4-1-2	7-4-5-6-3	16.34818	---	4.17%
7	1-4-5-8	2-5-8	4-5-2	7-4-1-2-3	16.63968	---	12.50%
8	1-2-5-8	2-5-8	4-1-2	7-4-5-2-3	17.68074	---	16.67%
9	1-2-5-8	2-5-8	4-1-2	7-4-5-6-3	16.34818	---	16.67%
10	1-2-5-8	2-5-8	4-5-2	7-4-5-6-3	15.42510	---	12.50%
11	1-2-5-8	2-5-8	4-5-2	7-4-1-2-3	17.68074	---	4.17%
12	1-4-7-8	2-5-8	4-5-2	7-4-5-6-3	16.34818	---	4.17%

Table 7 Summary of the simulation results on four OD pairs

Sequence of OD pairs	Results		Sequence of OD pairs	Results	
	with same initial condition	with different initial conditions		with same initial condition	with different initial conditions
A B C D	1	1	C A B D	3	10
A B D C	1	1	C A D B	2	10
A C B D	3	7	C B A D	5	11
A C D B	2	7	C B D A	4	7
A D B C	1	1	C D A B	2	10
A D C B	1	1	C D B A	4	12
B A C D	1	9	D A B C	1	8
B A D C	1	9	D A C B	1	8
B C A D	1	9	D B A C	1	8
B C D A	1	1	D B C A	1	1
B D A C	1	9	D C A B	1	8
B D C A	1	6	D C B A	1	1

Table 8 Shortest paths obtained for the different sequences of four OD pairs

Sequence of OD pairs		All zero initial condition for All OD pairs		Same initial condition for all OD pairs		Different initial condition for each OD pair	
1	1 -- 9	111255569		111444589		111222589	
2	8 -- 3	852222333	852222223	855555523	855555563	855555563	855555563
3	1 -- 9	111222569	111222569	111444569	111444589	111444589	111114589
4	8 -- 3	852222333	852222333	888555523	855555523	855555563	874444523
5	1 -- 9	111222569	111222569	111444569	111444589	111255569	144444589
6	8 -- 3	852222333	852222333	888555523	855555523	855552223	855555563
7	1 -- 9	111222569	111222569	111444569	111444589	111222569	111225589
8	8 -- 3	852222333	852222333	888555523	855555523	855555563	855555563
9	1 -- 9	111222569	111222569	111444569	111444589	111114589	111444589
10	8 -- 3	852222333	852222333	888555523	855555523	888555523	855555563

Table 9 Shortest paths from alternatingly presentation of two OD pairs in a 9-node network using different initial conditions. First column under each initial conditions corresponds to initialization with a different OD pair.

Percentage	Final Cost for Delay Cost Function Case	Final Cost for Derivative Delay Cost Function Case
1.0%	3.053078175	3.055994987
2.0%	3.046160793	3.042394257
4.0%	3.050959682	3.062650584
5.0%	3.071259595	3.066443539
10.0%	3.058737946	3.049192429
12.5%	3.092867184	3.107277489
16.7%	3.041032504	3.036917877
20.0%	3.070995522	3.111513518
25.0%	3.114019965	3.087734604
33.3%	3.138028240	3.095095252
50.0%	3.199145318	3.191892244
100.0%	3.507691004	3.420241559

Table 10 Final cost from a 9-Node Network with different flow increments

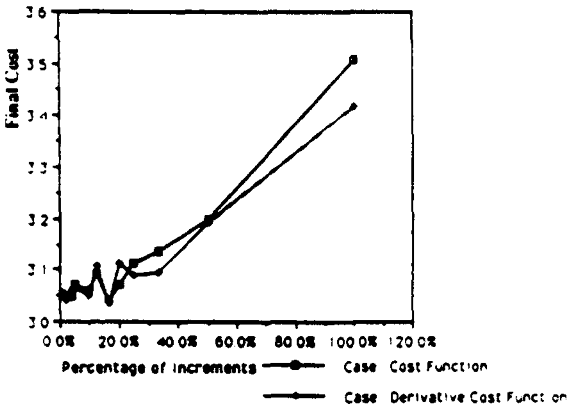


Figure 4 Final Cost vs. Percentage of Flow Increment

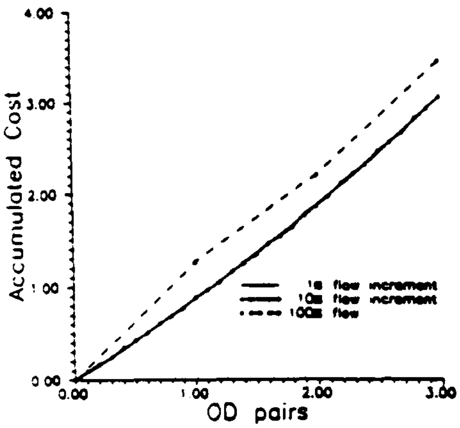


Figure 5 Accumulated Cost vs. OD pairs (Delay cost function case)

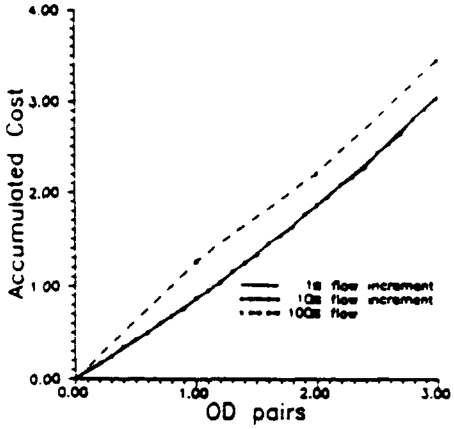


Figure 6 Accumulated Cost vs. OD pairs (Derivative Delay cost function case)

Table 11 Number of times a given path appeared in 1/(flow increment) trials using different flow increments. *Case 1: Delay Cost Function*

100% Flow Increment per Iteration

PAIR: 1

times path appeared	Flow	< Path >
1	0.1200000	9 8 5 4 1 0 0 0 0

PAIR: 2

	Flow	< Path >
1	0.1200000	2 1 4 7 0 0 0 0 0

PAIR: 3

	Flow	< Path >
1	0.1200000	8 5 2 3 0 0 0 0 0

The final cost is: 3.44939137

25% Flow Increment per Iteration

PAIR: 1

	Flow	< Path >
1	0.0300000	9 8 5 4 1 0 0 0 0
2	0.0600000	9 6 5 4 1 0 0 0 0
1	0.0300000	9 8 5 2 1 0 0 0 0

PAIR: 2

	Flow	< Path >
1	0.0300000	2 5 4 7 0 0 0 0 0
3	0.0900000	2 1 4 7 0 0 0 0 0

PAIR: 3

	Flow	< Path >
1	0.0300000	8 7 4 5 6 3 0 0 0
2	0.0600000	8 5 2 3 0 0 0 0 0
1	0.0300000	8 5 6 3 0 0 0 0 0

The final cost is: 3.10797119

1% Flow Increment per Iteration

PAIR: 1

	Flow	< Path >
9	0.0108000	9 8 5 2 1 0 0 0 0
12	0.0144000	9 8 5 4 1 0 0 0 0
16	0.0192000	9 6 5 2 1 0 0 0 0
55	0.0659998	9 6 5 4 1 0 0 0 0
1	0.0012000	9 8 5 4 1 4 1 0 0
5	0.0060000	9 6 5 4 1 4 1 0 0
1	0.0012000	9 8 7 4 1 0 0 0 0
1	0.0012000	9 6 9 6 5 4 1 0 0

PAIR: 2

	Flow	< Path >
28	0.0335999	2 1 4 7 0 0 0 0 0
63	0.0755995	2 5 4 7 0 0 0 0 0
9	0.0108000	2 5 8 7 0 0 0 0 0

PAIR: 3

	Flow	< Path >
46	0.0551999	8 5 6 3 0 0 0 0 0
49	0.0587999	8 5 2 3 0 0 0 0 0
2	0.0024000	8 7 4 5 2 3 0 0 0
2	0.0024000	8 7 4 5 6 3 0 0 0
1	0.0012000	8 7 4 1 2 3 0 0 0

The final cost is: 3.05268764

Table 12 Number of times a given path appeared in 1/(flow increment) trials using different flow increments. *Case 1: Derivative Delay Cost Function*

100% Flow Increment per Iteration

PAIR: 1

times path appeared	Flow	< Path >
1	0.1200000	9 8 5 4 1 0 0 0 0

PAIR: 2

	Flow	< Path >
1	0.1200000	2 1 4 7 0 0 0 0 0

PAIR: 3

	Flow	< Path >
1	0.1200000	8 5 6 3 0 0 0 0 0

The final cost is: 3.44939137

25% Flow Increment per Iteration

PAIR: 1

	Flow	< Path >
1	0.0300000	9 8 5 4 1 0 0 0 0
1	0.0300000	9 6 5 2 1 0 0 0 0
2	0.0600000	9 6 5 4 1 0 0 0 0

PAIR: 2

	Flow	< Path >
1	0.0300000	2 1 4 7 0 0 0 0 0
3	0.0900000	2 5 4 7 0 0 0 0 0

PAIR: 3

	Flow	< Path >
2	0.0600000	8 5 6 3 0 0 0 0 0
2	0.0600000	8 5 2 3 0 0 0 0 0

The final cost is: 3.07753086

1% Flow Increment per Iteration

PAIR: 1

	Flow	< Path >
2	0.0024000	9 8 5 2 1 0 0 0 0
69	0.0827993	9 6 5 4 1 0 0 0 0
10	0.0120000	9 8 5 4 1 0 0 0 0
17	0.0204000	9 6 5 2 1 0 0 0 0
2	0.0024000	9 8 7 4 1 0 0 0 0

PAIR: 2

	Flow	< Path >
52	0.0623999	2 5 4 7 0 0 0 0 0
7	0.0084000	2 5 8 7 0 0 0 0 0
41	0.0491999	2 1 4 7 0 0 0 0 0

PAIR: 3

	Flow	< Path >
37	0.0443999	8 5 6 3 0 0 0 0 0
56	0.0671998	8 5 2 3 0 0 0 0 0
5	0.0060000	8 7 4 5 2 3 0 0 0
2	0.0024000	8 7 4 1 2 3 0 0 0

The final cost is: 3.03548908

Table 13 Simulation results from a 16-Node Grid Network
with four OD-pairs: (1,8), (2,12), (14,4), (1,13)
Cumulative Results

OD-pair in order of presentation	*Initial Temperature*	*Temperature after 100 iterations*	*Final Energy after 200 iterations*	*Selected Path at the end of annealing*
(1,8)	250	30.70193	25.26316	1-2-3-7-8
(2,12)	250	30.70193	25.52632	2-6-7-11-12
(14,4)	250	30.70193	28.68826	14-10-6-7-3-4
(1,13)	250	30.70193	29.63563	1-5-9-13

The total energy is: 29.63563

Intermediate and final results for the fourth OD-pair (1,13)

Initial states for OD pair 4

1.000	0.040	0.111	0.097	0.086	0.092	0.083	0.079	0.049	0.091	0.086	0.002	0.019	0.092	0.092	0.000
0.000	0.004	0.036	0.115	0.041	0.047	0.116	0.057	0.049	0.096	0.100	0.061	0.101	0.118	0.043	0.000
0.000	0.112	0.056	0.111	0.005	0.018	0.034	0.100	0.043	0.004	0.049	0.008	0.105	0.114	0.047	0.000
0.000	0.019	0.118	0.097	0.010	0.029	0.093	0.041	0.101	0.120	0.023	0.108	0.078	0.085	0.082	0.000
0.000	0.072	0.002	0.001	0.111	0.087	0.055	0.071	0.072	0.065	0.017	0.042	0.007	0.020	0.067	0.000
0.000	0.105	0.020	0.093	0.007	0.024	0.115	0.036	0.118	0.084	0.104	0.056	0.082	0.086	0.093	0.000
0.000	0.119	0.040	0.032	0.088	0.043	0.084	0.014	0.034	0.054	0.102	0.007	0.012	0.057	0.069	0.000
0.000	0.010	0.107	0.120	0.004	0.069	0.098	0.108	0.015	0.040	0.043	0.022	0.032	0.043	0.117	0.000
0.000	0.051	0.025	0.025	0.067	0.070	0.103	0.093	0.089	0.038	0.115	0.123	0.055	0.080	0.083	0.000
0.000	0.097	0.073	0.086	0.017	0.108	0.017	0.092	0.013	0.079	0.040	0.016	0.002	0.039	0.049	0.000
0.000	0.039	0.087	0.073	0.080	0.008	0.106	0.122	0.113	0.124	0.085	0.034	0.078	0.022	0.124	0.000
0.000	0.121	0.036	0.025	0.112	0.061	0.091	0.087	0.007	0.076	0.086	0.038	0.037	0.048	0.068	0.000
0.000	0.080	0.029	0.070	0.019	0.071	0.005	0.012	0.027	0.002	0.109	0.059	0.076	0.006	0.008	1.000
0.000	0.102	0.070	0.118	0.018	0.001	0.078	0.059	0.023	0.042	0.109	0.087	0.002	0.043	0.093	0.000
0.000	0.119	0.053	0.121	0.052	0.081	0.038	0.007	0.081	0.097	0.040	0.117	0.074	0.073	0.124	0.000
0.000	0.065	0.105	0.002	0.119	0.028	0.100	0.098	0.086	0.082	0.021	0.015	0.004	0.101	0.004	0.000

iteration: 100 temp: 250.0000 pair: 4

1.000	0.205	0.119	0.074	0.071	0.070	0.072	0.070	0.060	0.065	0.064	0.054	0.044	0.006	0.004	0.000
0.000	0.011	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000
0.000	0.722	0.739	0.776	0.779	0.782	0.776	0.769	0.762	0.761	0.758	0.766	0.730	0.682	0.009	0.000
0.000	0.004	0.052	0.061	0.068	0.066	0.065	0.065	0.066	0.065	0.066	0.057	0.048	0.006	0.004	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.000	0.000	0.001	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000
0.000	0.004	0.050	0.059	0.065	0.064	0.064	0.064	0.066	0.068	0.074	0.082	0.134	0.218	0.824	0.000
0.000	0.000	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.012	0.001	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.001	0.001	0.001	0.011	0.026	1.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.011	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000

iteration: 200 temp: 30.70193 pair: 4

1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.985	0.996	0.996	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.994	0.981	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.928	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

The energy of the network after previous OD pairs: 29.63563

The final energy is: 29.63563

Model-Based Joint Detection/Estimation Approach for Multi-Sensor Data Fusion

Thomas W. Hilands

Applied Research Laboratory
The Pennsylvania State University
University Park, PA 16802

Stelios C. A. Thomopoulos

Decision and Control Systems Laboratory
Dept. of Electrical and Computer Engineering
The Pennsylvania State University
University Park, PA 16802

ABSTRACT

A nonlinear adaptive detector/estimator is introduced for single and multiple sensor data processing. The problem of target detection from returns of monostatic sensor(s) is formulated as a nonlinear joint detection/estimation (JDE) problem on the unknown parameters in the signal return. The unknown parameters involve the presence of the target, its range, and azimuth. The problems of detecting the target and estimating its parameters are considered jointly. A bank of spatially and temporally localized nonlinear filters is used to estimate the a *posteriori* likelihood of the existence of the target in a given space-time resolution cell. Within a given cell, the localized filters are used to produce refined spatial estimates of the target parameters. A decision logic is used to decide on the existence of a target within any given resolution cell based on the a *posteriori* estimates reduced from the likelihood functions. The inherent spatial and temporal referencing in this approach is used for automatic referencing required when multiple sensor data is fused together.

1. RANGE ESTIMATION FROM COLOCATED SENSORS

This section considers the problem of localizing a target in range space from data received at one or more colocated sensor(s). The range-Doppler space is partitioned into a number of resolution cells. Each cell is identified with a hypothesis that the signal is present in it. A JDE scheme is then used to localize the target and refine its parameter estimates. The measurements that are used to localize the target consist of signal returns corrupted by additive white Gaussian and non-Gaussian noise.

The problem is formulated using the JDE procedure adapted to problems with uncertain initial conditions[1-4]. The approach involves the operation of several nonlinear independent filters in parallel. In the case of Gaussian measurement noise the extended Kalman filter (EKF) is used for estimation. An extended high order filter (EHOF)[3,5] is used in non-Gaussian noise. The parallel filters are distinguished by the initial conditions used to set up the problem. Along with the state estimate the a *posteriori* probability of each hypothesis is computed recursively.

1.1 Problem Statement

Consider the problem of signal detection and parameter estimation in the context of the reception of an active echo return from a object that has been illuminated by a monostatic source. The situation is considered in which there are P collocated sources that illuminate the target simultaneously, but with different carrier frequencies designated ω_{p_c}. The received signal at each sensor is frequency-translated by mixing it with a signal at frequency ω_{p_t}. The resulting signal is low-pass filtered, and digitized at a rate f_s, which is at least twice the highest frequency in the data. The time between samples is denoted t_s. It is assumed that all sensors have the same digitization rate, and that all clocks are synchronized. The general expression for the received signal at the p^{th} sensor, under the signal-present assumption, can be written

$$z_{p_k} = a_{p_k}(\tau_k)p_{p_k}(\tau_k, \nu_k)r_{p_k}(\tau_k, \nu_k) + v_{p_k} \tag{1}$$

where $a_{p_k}(\tau_k)$ is the received signal amplitude, $p_{p_k}(\tau_k, \nu_k)$ is the pulse shaping function, and

$$r_{p_k}(\tau_k, \nu_k) = \cos\left[(\nu_k(\omega_{p_c}(k t_s - \tau_k))) - \omega_{p_c} k t_s\right] \tag{2}$$

v_{p_k} is white noise with $E[v_{p_k}] = 0$, $E[v_{p_k} v_{p_j}] = \sigma^2_{p_k}\delta(k-j)$, and τ_k is the time delay between signal transmission and reception. τ_k is a function of the range D_k between the receiver and the object, and is given by

$$\tau_k = \frac{2D_k}{c} \tag{3}$$

For unambiguous range estimation the uncertainty in τ_k, denoted $\Delta\tau_k$ is bounded by $\Delta\tau_k \leq 2\pi/(\nu_k\omega_{p_c})$. This is due to the fact that the cos(.) function is not monotonic (i.e. $r_{p_k}(\tau_1, \nu_k) = r_{p_k}(\tau_2, \nu_k)$, if $\tau_2 - \tau_1 = 2\pi/(\nu_k\omega_{p_c})$). $p_{p_k}(\tau_k, \nu_k)$ is the pulse shaping function, which has average energy E_p.

1.2 Joint Detection/Estimation

In this section we describe the JDE procedure for optimal estimation of time delay and Doppler shift assuming the presence of the target has been detected. The range of uncertainty in delay and Doppler is partitioned into a finite number of resolution cells. Each cell is associated with a hypothesis θ_i. The hypotheses are distinguished from each other by the initial conditions on the initial state estimates , $\hat{\mathbf{x}}_{0|0,\theta_i}$, and initial state covariances $P_{0|0,\theta_i}$. The measurement and process models are the same for each hypothesis. Let $\theta_i \in \Theta$ designate the parameter vector that describes the different initial conditions on the states. The parameter vector θ_i is also assumed to be time invariant. Under hypothesis H_{θ_i} the discrete time measurements are modeled according to

$$H_{\theta_i} \quad : \quad \mathbf{z}_k = \mathbf{g}_k(\mathbf{x}_k) + \mathbf{v}_k \quad \text{with i.c.'s } \hat{\mathbf{x}}_{0|0,\theta_i}, P_{0|0,\theta_i} \tag{4}$$

The measurement vector $\mathbf{z}_k$ is composed of the scalar measurements of the P individual sensors such that

$$\mathbf{z}_k = \left[z_{1_k} \; z_{2_k} \; \cdots \; z_{P_k}\right]^T \tag{5}$$

The state $\mathbf{x}_k$ is common for all $\theta_i \in \Theta$, and satisfies the discrete time process equation

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1} \tag{6}$$

The initial state estimate, the measurement noise, and the process noise are uncorrelated. The process and measurement noise are zero mean and distributed with covariances $E[\mathbf{w}_k\mathbf{w}_k^T] = Q_k$, and $E[\mathbf{v}_k\mathbf{v}_k^T] = R_k$.

For each $\theta_i \in \Theta$ (each assumed model), a minimum variance estimate of the model parameters is obtained recursively using the JDE technique. Using this technique a minimum variance estimate of the model parameters is obtained for every assumed model. These estimates are subsequently used to estimate the likelihood of each model being the correct one. Based on these likelihood estimates, a maximum a *posteriori* (MAP) decision criteria or a minimum mean squared error (MMSE) decision criteria can be used to select the proper model.

From Bayes' rule, the a *posteriori* probability of the parameter vector θ is updated recursively by[1-3]

$$P(\theta_i|\mathbf{Z}_k) = \frac{P(\theta_i|\mathbf{Z}_{k-1})\, p(\mathbf{z}_k|\mathbf{Z}_{k-1}, \theta_i)}{\sum_{m=1}^{M} P(\theta_m|\mathbf{Z}_{k-1})\, p(\mathbf{z}_k|\mathbf{Z}_{k-1}, \theta_m)} \tag{7}$$

where $\mathbf{Z}_{k-1} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots \mathbf{z}_{k-1}\}$. The initial condition for (7) is the a *priori* probability density function $p(\theta) \equiv p(\theta|\mathbf{Z}_0)$, which is assumed to be known. The densities $p(\mathbf{z}_k|\mathbf{Z}_{k-1}, \theta_i)$ are updated using the EKF[6] for estimation in Gaussian noise, or the EHOF[3,5] for estimation in non-Gaussian noise. Since the state vector $\mathbf{x}_k$ is common to all models, the minimum mean squared error (MMSE) estimate can be used. The MMSE estimate is expressed as a weighted average of the conditional state estmates $\hat{\mathbf{x}}_{k|k,\theta_i}$ over all θ_i as follows:

$$\hat{\mathbf{x}}^*_{k|k} = \sum_{i=1}^{M} P(\theta_i|\mathbf{Z}_k)\, \hat{\mathbf{x}}_{k|k,\theta_i}. \tag{8}$$

Model (4) can be extended to include the signal-absent case (null hypothesis) by augmenting the set of hypotheses $\{\theta_i\}$ with the null hypothesis θ_0 which has the associated noise-only measurement model

$$z_k = v_k, \tag{9}$$

and renormalization of the a *priori* distribution $P(\theta_i,\ i = 0, 1, \cdots, M$, where M is the number of resolution cells.

1.3 Specification of Initial Conditions

The localized initial conditions for each resolution cell are defined as follows: Let the time delay have mean $\hat{\tau}_0$ and density function $p_{\tau_0}(\tau_0)$. The distribution of τ_0 is segmented into N nonoverlapping segments such that the segment around some localized initial estimate $\hat{\tau}_{n_0}$ is defined by

$$p_{\tau_{n_0}}(\tau_{n_0}) = p_{\tau_0}(\tau_0) \qquad \alpha_n \leq \tau_0 \leq \alpha_{n+1} \qquad 1 \leq n \leq N \tag{10}$$

We have

$$\sum_{n=1}^{N} \int_{\alpha_n}^{\alpha_{n+1}} p_{\tau_{n_0}}(\tau)d\tau = \int_{-\infty}^{\infty} p_{\tau_0}(\tau)d\tau = 1$$

Define the scaling parameters ζ_n such that

$$\zeta_n \int_{\alpha_n}^{\alpha_{n+1}} p_{\tau_{n_0}}(\tau)d\tau = 1 \qquad 1 \leq n \leq N$$

Then the mean and variance of the initial conditions of the segmented model are given by

$$\hat{\tau}_{n_0} = E[\tau_{n_0}] = \zeta_n \int_{\alpha_n}^{\alpha_{n+1}} \tau p_{\tau_{n_0}}(\tau)d\tau$$

$$\mathrm{Var}[\tau_{n_0}] = \zeta_n \int_{\alpha_n}^{\alpha_{n+1}} \tau^2 p_{\tau_{n_0}}(\tau)d\tau - \hat{\tau}_{n_0}^2$$

With N different initial conditions on τ_0 there are N different resolution cells for referencing the measurements. A different filter is initialized in each resolution cell. The total number of cells in the resolution space can be large, *depending* on the desired accuracy in the parameter resolution. However, the filters can be run in parallel, and independent of each other, thus reducing the execution time to that of a single filter.

The parameter vector θ_i, $1 \leq i \leq N$, is defined to be the i^{th} resolution cell and is used to define N initial conditions on the state variables τ. The a *priori* probabilities of each hypothesis are determined by integrating the density function $p_{\tau_0}(\tau_0)$ and over the limits defined for each hypothesis. They are given by

$$P(\theta_i) = \int_{\alpha_n}^{\alpha_{n+1}} p_{\tau_0}(\tau)d\tau \tag{11}$$

1.4 Joint Detection/Estimation of Time Delay

This section addresses the model in which the state $x_k = \tau_k$ is unknown and to be estimated. The parameter vector θ_i is defined as before. Hypothesis H_i is now given by

$$H_i \quad : \quad z_{p_k} = \begin{cases} v_k & k\,t_s < \hat{\tau}_k \\ g_{p_k}(\hat{\tau}_k) + v_k & \hat{\tau}_k \leq k\,t_s < \hat{\tau}_k + t_w \\ v_k & k\,t_s \geq \hat{\tau}_k + t_w \end{cases} \tag{12}$$

with initial conditions

$$\begin{aligned} \hat{x}_{0|0,\theta_i} &= [\hat{\tau}_{n_0}]^T \\ P_{0|0,\theta_i} &= \left[\mathrm{Var}[\tau_{n_0}]\right] \end{aligned} \tag{13}$$

where

$$s_{p_k}(\hat{\tau}_k) = a_{p_k}(\hat{\tau}_k)p_{p_k}(\hat{\tau}_k)r_{p_k}(\hat{\tau}_k) \tag{14}$$

$$\begin{aligned} a_{p_k}(\hat{\tau}_k) &= \frac{4A_{p_1}}{(c\,\hat{\tau}_k)^2} \\ p_{p_k}(\hat{\tau}_k) &= 0.5\,(1-\cos(2\pi\,\nu_{m_0}(k\,t_s - \hat{\tau}_k)/t_w)) \\ r_{p_k}(\hat{\tau}_k, \nu_{m_0}) &= \cos\left[(\nu_{m_0}(\omega_{p_c}(k\,t_s - \hat{\tau}_k))) - \omega_{p_1} k\,t_s\right] \end{aligned} \tag{15}$$

where it is observed that the amplitude function $a_{p_k}(.)$ reflects the transmitted amplitude A attenuated by sperical spreading loss.

1.5 Experimental Evaluation

Both single and double sensor models ($P = 1$, and $P = 2$) in (5) were selected for experimental evaluation. For this evaluation the sampling frequency was $f_s = 100 \times 10^6$ Hz, the pulse width was set to $12\,t_s$ and c, the speed of propagation, was 186000 miles/sec. For all tests, the nominal time delay and Doppler were $\tau_{\text{non}} = 0.000324$ and $(\nu_{\text{nom}} - 1) = 8.96 \times 10^{-7}$ respectively, corresponding to a target at a nominal range of 10 miles. traveling at 300 mph Doppler velocity.

It was assumed that the error in the time delay estimate was uniformly distributed at $\pm 3.5\,t_s$ about the nominal delay. The corresponding variance is then $(7\,t_s)^2/12$. The error in the Doppler estimate was assumed to be uniformly distributed at $\pm 7.47 \times 10^{-7}$ about the nominal Doppler. This corresponds to an error in the Doppler velocity of ± 250 mph. The corresponding variance is 1.85×10^{-13}.

1.5.1 Single Sensor Evaluation

The single sensor model was used to compare the use of multiple filters ($N = 7$) to a single filter ($N = 1$) for JDE. With only one filter, $\hat{x}_{0|0,\theta_1} = \hat{\tau}_{\text{nom}}$, $P_{0|0,\theta_1} = (7\,t_s)^2/12$, as described previously. The initial estimates of time delay for the multiple filter implementation are given by $\hat{\tau}_{n_0} = (n-4) * t_s + \tau_{\text{nom}}$, $n = 1, 2, \cdots 7$. Thus, the initial delay estimates were separated by t_s, with $\text{Var}(\tau_{n_0}) = t_s^2/12$, $\forall n$. The *a priori* probabilities are given by $P(\theta_n|\mathbf{Z}_0) = 1/N$, $1 \le n \le N$.

The Monte Carlo simulation results for JDE with a single filter ($N = 1$) and a bank of seven filters ($N = 7$) are shown in Figure 1(a). In this figure the mean squared error (MSE) of the estimation error in τ_k is shown as a function of SNR, where SNR $\equiv 10\log(E_s/\sigma_n^2)$, for $\tau_k \le k\,t_s < \tau_k + t_w$, and E_s is the average received signal energy per sample. Each point on the graph represents the results of 500 simulation runs. Both the MAP and MMSE estimates are shown in Figure 1(a). The MAP and MMSE estimates are the same for $N = 1$. Also shown on this graph are the results for the detection-only (D-O) technique, which is implemented by fixing the estimates at their initial values. The noise is Gaussian, and the EKF is used to perform estimation in the JDE method. The JDE ($N = 7$) implementation gives better results than the D-O method, particularly at higher SNR. This is expected since the filter in the JDE method allows a considerable refinement estimates at higher SNR as compared to low SNR where the larger noise covariance restricts the filter gain. At -5 dB SNR the JDE and D-O implementations perform identically. In general, the MMSE estimates are better than the MAP estimates, particularly at low SNR's. The JDE ($N = 1$) implementation gives the worst overall performance. The filter used in this implementation often converges to poor final estimates due to the tendency, mentioned previously, of time delay to converge to values that are separated from the actual time delay by multiples of $\pm 1/f_c$.

The JDE ($N = 7$) technique is evaluated in lognormal noise in Figure 1(b) for the single sensor model. The MMSE estimates of τ_k are shown in this figure for the EKF and for the EHOF. The EKF is evaluated in two configurations. In the first configuration, the Gaussian pdf is used to evaluate the detection statistic given by equation (7). In the second configuration, the lognormal pdf is used. The EHOF is evaluated using the lognormal pdf only. The EKF in the second configuration and the EHOF give very similar results at low SNR. However, at high SNR the EHOF outperforms the EKF. When the Gaussian pdf is used in conjunction with the EKF to localize the target, the results are significantly worse than when the proper lognormal pdf is used. This advantage is particularly evident at low SNR's.

1.5.2 Double Sensor Evaluation

In the multiple sensor case ($P > 1$), the sensors may have different carrier frequencies (ω_{p_c}), and different translation frequencies (ω_{p_t}). A two-sensor ($P = 2$) model was evaluated in which $\omega_{c_1} = 2\pi * 10 \times 10^6$, $\omega_{c_2} = 2\pi * 30 \times 10^6$, and $\omega_{t t_1} = \omega_{t_2} = 0$. The MMSE results of this evaluation for JDE ($N = 7$) are given in Figure 1(c). The single-sensor ($P = 1$) MMSE results are also shown in this figure. This figure illustrates the distinct advantage of centralized fusion for JDE.

1.5.3 Multiple Pulse Processing

The results of processing two pulses are given in Figure 1(d). The EKF and EHOF are configured such that the initial error covariance is reset at the beginning of each pulse. The rationale for this is to re-excite the system. This helps to allow poor estimates to possibly converge to smaller errors, and it has been shown experimentally[3], that it does not significantly effect those estimates that have already converged close to the actual state value. Figure 1(d) shows an improvement of about 3 dB for the two pulse estimate over the single pulse estimate.

2. RANGE AND AZIMUTH ESTIMATION FROM NONCOLOCATED SENSORS

Consider the situation of two spatially separated sensors, $s1$ and $s2$. Each of the two sensors attempts to detect and track objects coming into its respective area of coverage. The coverage of the two sensors is assumed to overlap in space, but not entirely. The sensor geometry is shown in Figure 2. In the overlap region the data received by the two sensors can be combined to get a more accurate estimate of target parameters or to estimate parameters that cannot be estimated with one sensor alone. In the overlap region the estimates from the individual sensors are combined to form improved target parameter estimates. We consider the case where each of the sensors may have different types of tracking devices such as optical trackers, various types of radars, etc. It is assumed that these sensors transmit a signal and process the echo returned from that signal. The signals are corrupted by additive Gaussian noise due to thermal effects within the receiver, and by clutter which may be due to non-Gaussian distortion such as sea clutter or other multipath spreading. Typical distributions used to model this distortion include the Rayleigh, Weibull or lognormal distributions[7]. The thermal noise at the receiver is assumed to be uncorrelated from sensor to sensor.

2.6 System Model

Assume that each sensor consists of a phased array or some other sensing device that can produce target angle estimates along with estimates of time delay and Doppler shift. It is assumed that there are two separate measurements taken at each sensor - one measurement at each of the offset phase centers. The received signal at the p^{th} sensor may be described by

$$\mathbf{z}_{p_k} = \mathbf{g}_{p_k} + \mathbf{u}_{p_k} + \mathbf{v}_{p_k} \tag{16}$$

where $\mathbf{g}_{p_k}$ represents the received signal, $\mathbf{u}_{p_k}$ is the clutter, and $\mathbf{v}_{p_k}$ is the Gaussian noise at the k^{th} sampling interval. Since there are two measurements observed at each sensor, the received signal can be more explicitly expressed as

$$\begin{bmatrix} z_{p1_k} \\ z_{p2_k} \end{bmatrix} = \begin{bmatrix} g_{p1_k} \\ g_{p2_k} \end{bmatrix} + \begin{bmatrix} u_{p1_k} \\ u_{p2_k} \end{bmatrix} + \begin{bmatrix} v_{p1_k} \\ v_{p2_k} \end{bmatrix} \tag{17}$$

Two unknown delays, τ_{p1} and τ_{p2}, are introduced in the received signal $\mathbf{g}_{p_k}$. The delay τ_{p1} is the round-trip propagation time from the center of the sensor to the target and back to the sensor. Referring to Figure 3, this is the time for the signal to travel from point P_p to O and back to point P_p. From τ_{p1} the range to the target can be determined using the relationship

$$D_p = \frac{\tau_{p1}}{2c} \tag{18}$$

where c is the speed of propagation. The delay τ_{p2} is the difference in time for the signal to reach from point P_{p1} to point P_{p2}. The difference in the propagation distance is given by $c\,\tau_{p2}$. The differential angle $\Delta\phi_p$ to the target from sensor p, which represents the difference between the sensor pointing angle ϕ_{p_0} and the actual target angle ϕ_p, is then

$$\begin{aligned}\Delta\phi_p &= \sin^{-1}\left(\frac{c\,\tau_{p2}}{d_p}\right)\\ \phi_p &= \phi_{p_0} + \Delta\phi_p\end{aligned} \tag{19}$$

where d_p is the distance between the two offset phase centers in the phased array for sensor p.

2.6.1 Single Observer Model

Using estimates of τ_{p1} and τ_{p2} from one sensor the target position can be estimated through the relations (18, and 19). Define the state variable vector for sensor p as $\mathbf{x}_{p_k} = \left[\tau_{p1_k}\ \tau_{p2_k}\right]^T$. It is assumed that the state does not change while the pulse is being reflected from it. Therefore the process dynamics are zero; that is, the state transition matrix is unity and there is no process noise. In terms of the state variables the received signal at the p^{th} sensor is

$$\mathbf{g}_{p_k} = \begin{bmatrix} g_{p1_k} \\ g_{p2_k} \end{bmatrix} = \begin{bmatrix} a_{p1_k}(\mathbf{x}_{p_k}) p_{p1_k}(\mathbf{x}_{p_k}) r_{p1_k}(\mathbf{x}_{p_k}) \\ a_{p2_k}(\mathbf{x}_{p_k}) p_{p2_k}(\mathbf{x}_{p_k}) r_{p2_k}(\mathbf{x}_{p_k}) \end{bmatrix} \tag{20}$$

where

$$\begin{aligned} a_{pj_k}(\mathbf{x}_{p_k}) &= \frac{4A}{(c\,(x_{p_k}(1) - \kappa_j x_{p_k}(2)/2))^2} \\ p_{pj_k}(\mathbf{x}_{p_k}) &= 0.5 * (1 - \cos(2\pi\nu_p(kt_s - x_{p_k}(1) + \kappa_j x_{p_k}(2)/2)/t_{w_p})) \\ r_{pj_k} &= \cos(\nu_p(\omega_p(kt_s - x_{p_k}(1) + \kappa_j x_{p_k}(2)/2))) \end{aligned} \tag{21}$$

for $j = 1, 2$. $\kappa_j = +1$ whenever $j = 1$. $\kappa_j = -1$ whenever $j = 2$. ν_p is the doppler velocity (assumed known in this case), A is the transmitted amplitude, and $a_{pj_k}(.)$ reflects attenuation due to spherical spreading loss. The definition of $p_{pj_k}(.)$ given above represents the Hanning pulse type with pulse width t_{w_p}. The EKF equations for the constant state model given above are given by

$$\begin{aligned} K_{p_k} &= P_{p_{k-1|k-1}} H^T_{p_k} (G_{p_k} P_{p_{k-1|k-1}} G^T_{p_k} + R^{(2)}_{p_k})^{-1} \\ P_{p_{k|k}} &= (I_n - K_{p_k} G_{p_k}) P_{p_{k-1|k-1}} \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k-1|k-1} + K_{p_k} \tilde{\mathbf{z}}_{p_k} \\ \tilde{\mathbf{z}}_{p_k} &= \mathbf{z}_{p_k} - \mathbf{g}_{p_k}(\hat{\mathbf{x}}_{k|k-1}) \end{aligned} \tag{22}$$

where $R^{(2)}_{p_k}$ is the measurement covariance, K_{p_k} is the filter gain, and G_{p_k} is the Jacobian of the measurement model[3,5]. The EHOF incorporates 3^{rd} and 4^{th} order estimation error and measurement error moments. However, the equations are very lengthy and are not presented here.

2.6.2 Double Observer Model

When information is available from two sensors, that is, whenever the target is in the overlap region, and the target is illuminated simultaneously by the two radars, the Doppler and time delay estimates from each sensor can be combined to obtain a better estimate of target position and velocity.

Let X' and Y' denote the directions of a local coordinate system as shown in the insert in Figure 2. Let ϕ_{1_0} and ϕ_{2_0}, the pointing angles of the two sensors, be chosen such that $\phi_{2_0} - \phi_{1_0} = 90\,\text{deg}$. In this case the direction X' points directly along the line of sight (LOS) of s_2, and perpendicular to the LOS of s_1. Likewise, Y' points directly along the LOS of s_1 and perpendicular to the LOS of s_2. X' is the in-track direction for s_1 and the cross-track direction for s_2. Y' is it in-track direction for s_2 and the cross-track direction for s_1. For small angles $\Delta\phi_p$ such that $\sin(\Delta\phi_p \approx 0)$, the position estimates in the X', Y' coordinate system, which can be found from either sensor, are given by

$$\begin{aligned}
\hat{O}_{x'_1} &= D_{1_0} c\hat{\tau}_{12}/d_1 \\
\hat{O}_{x'_2} &= -(c\hat{\tau}_{21}/2 \cdot\cdot D_{2_0}) \\
\hat{O}_{y'_1} &= (c\hat{\tau}_{11}/2 - D_{1_0}) \\
\hat{O}_{y'_2} &= D_{2_0} c\hat{\tau}_{22}/d_2
\end{aligned} \tag{23}$$

where D_{p_0} is the nominal range from sensor p to the center of the insert in Figure 2. The associated position error variances are given by

$$\begin{aligned}
\sigma^2_{x'_1} &= D^2_{1_0} c^2 \mathrm{Var}[\tau_{12}]/d_1^2 \\
\sigma^2_{x'_2} &= c^2 \mathrm{Var}[\tau_{21}]/4 \\
\sigma^2_{y'_1} &= c^2 \mathrm{Var}[\tau_{11}]/4 \\
\sigma^2_{y'_2} &= D^2_{2_0} c^2 \mathrm{Var}[\tau_{22}]/d_2^2
\end{aligned} \tag{24}$$

If it is assumed that the time delay estimation errors have Gaussian distributions, then the maximum likelihood estimates of the target position in the overlap region D_2, which are the weighted sums of the estimates at each sensor, are given by

$$\hat{O}_{x'} = \frac{\sigma^2_{x'_2} D_{1_0} c\hat{\tau}_{12}/d_1 - \sigma^2_{x'_1}(c\hat{\tau}_{21}/2 - D_{2_0})}{\sigma^2_{x'_1}\sigma^2_{x'_2}} \tag{25}$$

$$\hat{O}_{y'} = \frac{\sigma^2_{y'_2}(c\hat{\tau}_{11}/2 - D_{1_0}) + \sigma^2_{y'_1} D_{2_0} c\hat{\tau}_{22}/d_1}{\sigma^2_{y'_1}\sigma^2_{y'_2}} \tag{26}$$

2.7 Joint Detection/Estimation

The target search region has been localized to the rectangular box shown in Figure 2. This box is subdivided into several resolution cells as shown in this figure. The beam pattern from sensor s_1 allows this sensor to detect a target and estimate its parameters if the target is located in resolution cells 1 through 21. Sensor s_2 can detect the target if it is in cells 11 through 15, 22 through 25, or 26 through 31. If the target is not located in any of these cells then the target is declared not present (or more precisely, not detectable) . This situation is represented by the null hypothesis H_0. The resolution cells are grouped into regions which will be used for minimum mean squared error estimation. If the target is located in regions R_1 (resolution cells 1 through 9) or R_3 (resolution cells 16 through 21) only sensor s_1 can detect the target. Regions R_4 (resolution cells 22 through 25) and R_5 (resolution cells 26 through 31) correspond to the coverage area of sensor s_2 only. If the target is located in region R_2 (resolution cells 10 through 15) both sensors can detect the target and perform parameter estimation. The remaining area in the rectangle in Figure 2is designated as region R_0, where neither sensor can detect the target.

Let $\theta_i \in \Theta$ designate the parameter vector that describes the different combination of model uncertainty and initial condition uncertainty. The parameter vector θ_i is assumed to be time invariant. The parameter vector θ_i, $1 \leq i \leq 56$ is defined to be the i^{th} resolution cell and is used to define 56 different combinations initial conditions and models. i corresponds to the range resolution cell number determined from the initial conditions on the two time delays from each sensor.

In general, hypothesis H_i, representing the hypothesis that the target is located in resolution cell i, is defined by

$$H_i : \begin{cases} \mathbf{z}1_k = \mathbf{g}1_k + \mathbf{u}1_k + \mathbf{v}1_k \\ \mathbf{z}2_k = \mathbf{g}2_k + \mathbf{u}2_k + \mathbf{v}2_k \end{cases} \tag{27}$$

where u_{j_k} and v_{j_k}, $j = 1, 2$, represent the non-Gaussian and Gaussian noise, respectively, present at the p^{th} sensor.

In regions R_1, R_2, and R_3, where sensor s_1 can detect the target, the component g_{1m_k} is defined by (20) as

$$g_{1m_k} = \begin{cases} a_{1m_k}(.)p_{1m_k}(.)r_{1m_k}(.) & \hat{\tau}_{1m_{i_k}} \leq k\,t_s < \hat{\tau}_{1m_{i_k}} + t_{w_1} \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

In the regions R_0, R_4 and R_5, $g_{1m_k} = 0$, $\forall\, k$, $m = 1, 2$. The delay $\hat{\tau}_{pm_{i_k}}$ is given by

$$\hat{\tau}_{pm_{i_k}} = \hat{\tau}_{p1_{i_k}} + \kappa_m \hat{\tau}_{p2_{i_k}} \tag{29}$$

where $\kappa_m = +1$ whenever $m = 1$, and $\kappa_m = -1$ whenever $m = 2$. In regions R_2, R_4, and R_5, where sensor s_2 can detect the target, the component g_{2m_k} is defined by (20) as

$$g_{2m_k} = \begin{cases} a_{2m_k}(.)p_{2m_k}(.)r_{2m_k}(.) & \hat{\tau}_{2m_{i_k}} \leq k\,t_s < \hat{\tau}_{2m_{i_k}} + t_{w_2} \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

In the regions R_0, R_1 and R_3, $g_{2m_k} = 0$, $\forall\, k$, $m = 1, 2$.

The initial conditions are given by

$$\begin{aligned} \hat{x}_{p_{0|0,\theta_i}} &= [\hat{\tau}_{p1_{i_0}}, \hat{\tau}_{p2_{i_0}}]^T \\ P_{p_{0|0,\theta_i}} &= \mathrm{Diag}\left[\mathrm{Var}[\hat{\tau}_{p1_{i_0}}], \mathrm{Var}[\hat{\tau}_{p2_{i_0}}],\right] \end{aligned} \tag{31}$$

The initial estimates $\hat{\tau}_{p1_{i_0}}, \hat{\tau}_{p2_{i_0}}, p = 1, 2$ are chosen such that the position of the target for a signal received ate sensor p is at the center of resolution cell i. The variances $\mathrm{Var}[\hat{\tau}_{p1_{i_0}}]$ and $\mathrm{Var}[\hat{\tau}_{p2_{i_0}}]$ are determined based on a uniform distribution of the error within the cell.

Define $\mathbf{Z}_k = [\mathbf{z}_1, \mathbf{z}_2, \cdots \mathbf{z}_k]$, where $\mathbf{z}_k = [\mathbf{z}_{1_k}^T, \mathbf{z}_{2_k}^T]^T$, as the set of all measurements up to time k, and let $p(\mathbf{z}_k|\mathbf{Z}_{k-1}, \theta_i)$ be the probability density function of $\mathbf{z}_k$ given the measurements $\mathbf{Z}_{k-1}$ and hypothesis H_i. The *a posteriori* probability of hypothesis H_i is given by

$$P(\theta_i|\mathbf{Z}_k) = \frac{P(\theta_i|\mathbf{Z}_{k-1})\,\Lambda_i(\mathbf{z}_k)}{\sum_{m=0}^{N} P(\theta_m|\mathbf{Z}_{k-1})\,\Lambda_m(\mathbf{z}_k)} \tag{32}$$

where $\Lambda_i(\mathbf{z}_k)$ is the likelihood ratio defined by

$$\Lambda_k(\mathbf{z}_k) = \frac{p(\mathbf{z}_k|\mathbf{Z}_{k-1}, \theta_k)}{p(\mathbf{z}_k|\mathbf{Z}_{k-1}, \theta_0)} \tag{33}$$

The minimum mean squared error estimate can be found be combining the estimates from all of the cells with a particular region. If the state vector $\mathbf{x}_k$ is common to all models the minimum mean squared error (MMSE) estimate can be used. The MMSE estimate for sensor p in region R_r can be expressed by

$$\hat{\mathbf{x}}^{*}_{p_{k|k}} = \sum_{\text{cell}\,i \in R_r} P(\theta_i|\mathbf{Z}_k)\,\hat{\mathbf{x}}_{p_{k|k,\theta_i}}. \tag{34}$$

The most likely region is selected using the MAP criterion. Define as the hypothesis that the target is located in region R_r as I_r, $r = 0, 1, \cdots, 5$. The *a posteriori* probability associated with region R_r is the sum of the *a posteriori* probabilities of all of the cells in that region. This region-level probability is given by

$$P(I_r|\mathbf{Z}_k) = \sum_{\text{cell}\,i \in R_r} P(\theta_i|\mathbf{Z}_k) \tag{35}$$

The most likely region is chosen according to

$$\text{Choose} \quad I_r \quad : \quad r = \mathrm{argmax}_{r=0,\cdots,5,\theta_i \in \Theta}\, P(I_r|\mathbf{Z}_k) \tag{36}$$

2.7.1 Definition of Priors

The *a priori* probabilities of each hypothesis are based on the area coverage of the sensors. The total number of resolution cells shown in Figure 2 is 56. Of these, 25 are located in region R_0. All cells are assumed to have equal probability of containing the target. The *a priori* probabilities are given by $P(\theta_0) = 25/56$, $P(\theta_i) = 1/56$, $i = 1, 2, \cdots 31$. The probabilities associated with regions R_r, $r = 0, 1, \cdots, 5$ are given by $P(I_0) = 25/56$, $P(I_1) = 9/56$, $P(I_2) = 5/56$, $P(I_3) = 6/56$, $P(I_4) = 4/56$, $P(I_5) = 6/56$.

2.8 Simulation Experiments

An experimental study was conducted to evaluate the performance of the multi-sensor fusion technique. In this evaluation the measurement noise consisted of 50% Lognormal Noise and 50% Gaussian noise. The nominal angles from sensors s_1 and s_2 to the target were $\phi_{1_0} = 45$ deg and $\phi_{2_0} = 135$ deg, respectively. The nominal range from s_1 to the target was $D_1 = 10$ miles. The nominal range from sensor s_2 to the target D_2 was chosen such that the received signal at s_2 was 5 dB higher than at s_1 for the same transmitted signal level and target strength.

The carrier frequencies used by the two sensors were the same at $f_c = 10 \times 10^6$. Both sensors sample the signal at a rate $f_s = 100 \times 10^6$, and both signals have the same pulse width $t_{w_p} = 12/f_s$, $p = 1, 2$. The resolution cell width is $1/f_s$ seconds. The associated initial error variance on time delays τ_{11_0} and τ_{21_0} is $t_s^2/12$. The corresponding range resolution cell width is $\Delta r_p = c/(2f_s)$. Thus, the initial variance for the angle-measurement delays is (19) $\mathrm{Var}[\tau_{12_0}] = ((d_p c)/(2 f_s D_p))^2/12$, $p = 1, 2$. d_p, the separation between phase centers at the sensor was chosen to be 3 feet for each sensor. Simulations were performed for SNR's (at sensor s_1) ranging from -10dB to 10dB. 500 random target positions were chosen at each SNR. Of these 500 trials, 228 target positions randomly chosen in region R_0, 91 in R_1, 54 in R_2, 44 in R_3, 40 in R_4, and 40 in R_5. The results given here are for monopulse processing(i.e. one pulse repetition interval (PRI)).

The probabilities of missed detection $P(I_0|I_r)$ and correct classification (i.e. not only detection of the target but correct localization at the region level) $P(I_r|I_r)$, $r = 1, \cdots, 5$ are displayed in Table 1. The probability of misclassification, which is not shown in this table, is given by $P(I_q|I_r) = 1 - P(I_r|I_r) - P(I_0|I_r)$, $q \neq r$. Sensor s_2 outperforms sensor s_1, which is to be expected since the SNR at s_1 is 5 dB higher than the SNR at sensor s_2. In the overlap region, R_2, the classification performance is better than in any other region, with an 85% probability of correct classification at -10 dB SNR. Additional numerical results have been generated[3,5] with complete probability of detection (PD) and probability of false alarm (PFA). What appears as a discrepancy in $P(I_r|I_r)$ at -5 dB SNR for $r = 2, 3, 4$ is due to statistical error due to small sample size.

Table 1. Probabilities of Missed Detection and Correct Classification - Region Level

SNR(dB)		Probability				
		$r=1$	$r=2$	$r=3$	$r=4$	$r=5$
-10	$P(I_0\|I_r)$	0.35	0.074	0.50	0.15	0.16
	$P(I_r\|I_r)$	0.57	0.85	0.45	0.78	0.79
-5	$P(I_0\|I_r)$	0.13	0.019	0.23	0.025	0.023
	$P(I_r\|I_r)$	0.87	0.96	0.77	0.98	0.98
0	$P(I_0\|I_r)$	0.022	0.0	0.023	0.0	0.0
	$P(I_r\|I_r)$	0.98	1.0	0.98	1.0	1.0
5	$P(I_0\|I_r)$	0.0	0.0	0.0	0.0	0.0
	$P(I_r\|I_r)$	1.0	1.0	1.0	1.0	1.0
10	$P(I_0\|I_r)$	0.0	0.0	0.0	0.0	0.0
	$P(I_r\|I_r)$	1.0	1.0	1.0	1.0	1.0

The estimation results are shown in Figure 4. All results shown in this figure are in reference to the (X', Y') coordinate system. Figure 4(a) shows the average mean squared error for those detections in regions R_1 and R_3, in which only s_1 has coverage. Figure 4(c) shows similar results for regions R_4 and R_5, which are covered by sensor s_2. Figure 4(c) also illustrates the 5 dB performance for sensor s_2 over that for s_1. Figure 4(b) shows the results for both sensors in region R_2. In this region, as shown in Table 3 the proper cell is almost always found. Thus the cross-range estimation error variance should improve by about 6 dB (20log(2)) for sensor s_2, since the cross-range error for s_2 has been localized from 2 cells down to 1. Similarly, the cross-range error variance for sensor s_1 in Region R_2 is reduced by about 10 dB (20log(3)) since the target has been localized from 3 cells down to 1. This improvement is evident in Figure 4(b). Figure 4(d) shows the estimation results using the combined measurents obtained from (25, 26). Because of the larger variance in the cross-range error for each sensor and the fact that the intersection of the LOS's between the two sensors are perpendicular, the combined estimate consists of the X' estimate from sensor s_2 and the Y' estimate from sensor s_1.

3. CONCLUSION

A model-based adaptive detection/estimation approach has been presented for multi-sensor fusion. It is shown that excellent performance can be obtained for both target detection and target parameter estimation using this technique. A significant advantage of this technique is that each sensor can perform detection and parameter estimation in a decentralized mode. The final estimates and a *posteriori* probabilities from each sensor are processed by a centralized processor to derive the optimum estimate. The method provides an automatic referencing mechanism of the data from the different sensors (automatic data alignment) as long as the geometry and timing of the sweeping beams are known. For optimal target resolution performance, it is found that the lines of sight of the two sensors should be perpendicular to each other at any given time, requiring special synchronization.

4. REFERENCES

1 A. Fredriksen, D. Middleton, and D. Vanderlinde,, "Simultaneous Signal detection and Estimation Under Multiple Hypotheses," *IEEE Trans. Info. Theory*, vol. IT-18, no. 5, pp. 607–614, September, 1972.

2 D. G. Lainiotis, "Optimal Adaptive Estimation : Structure and Parameter Adaptation," *IEEE Trans. Automatic Control*, vol. AC-16, no. 2, pp. 160–170, April, 1971.

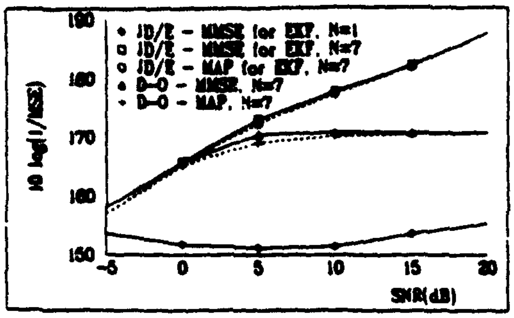
3 T. W. Hilands, "High Order Nonlinear Estimation with Signal Processing Applications," *Ph. D. Thesis*, The Pennsylvania State University, 1992.

4 S. C. A. Thomopoulos and T. W. Hilands, "Nonlinear Adaptive Detection/Estimation for Single and Multiple Radar Processing," *IEEE MILCOM '92*, October 11-14, 1992, San Diego, CA.

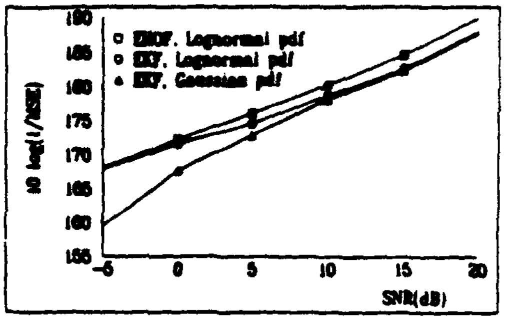
5 T. W. Hilands and S. C. A. Thomopoulos , "High Order Filters for Estimation in Non-Gaussian Noise," *Canadian Conference on Electrical and Computer Engineering, CCECE-92*, Toronto, Canada, Sept. 13-16, 1992.

6 A. Gelb (Ed), *Applied Optimal Estimation*, MIT Press, 1974.

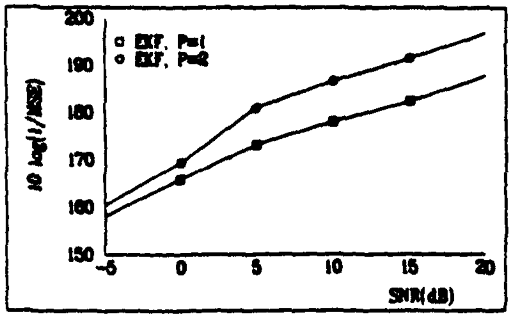
7 M. I. Skolnik, *Introduction to Radar Systems*, 2nd ed., McGraw-Hill, New York, 1980, pp. 478 – 479.



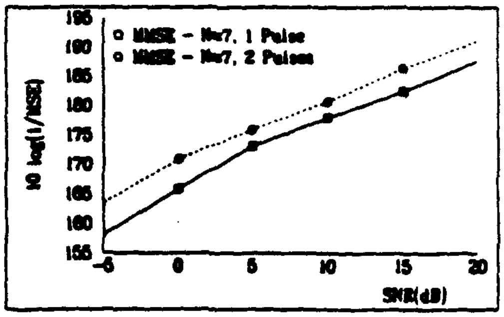
(a) Error in $\hat{\tau}_2$, Gaussian Noise, $P = 1$



(c) JD/E - MMSE for $\hat{\tau}_2$, Lognormal Noise, $N = 7$, $P = 1$



(b) JD/E - MMSE for $\hat{\tau}_2$, $N = 7$, Multiple Sensor



(d) JD/E - MMSE for $\hat{\tau}_2$, $N = 7$, Multiple Pulses

Figure 1   JD/E Performance for Time Delay Estimation

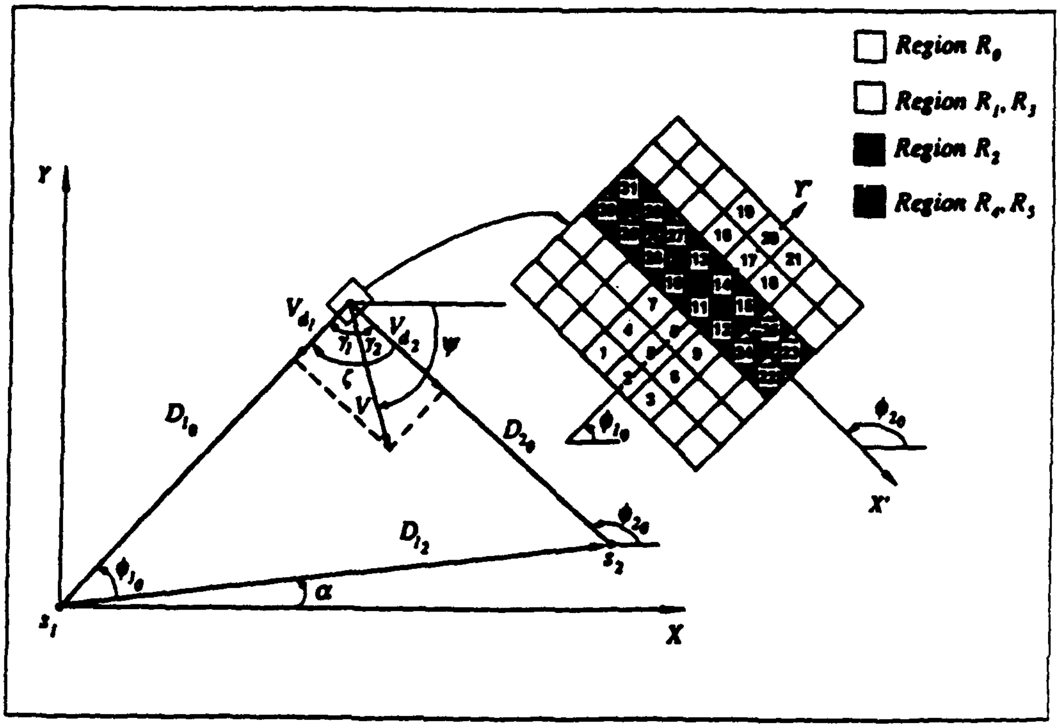


Figure 2   Sensor/Target Geometry for Multisensor Fusion

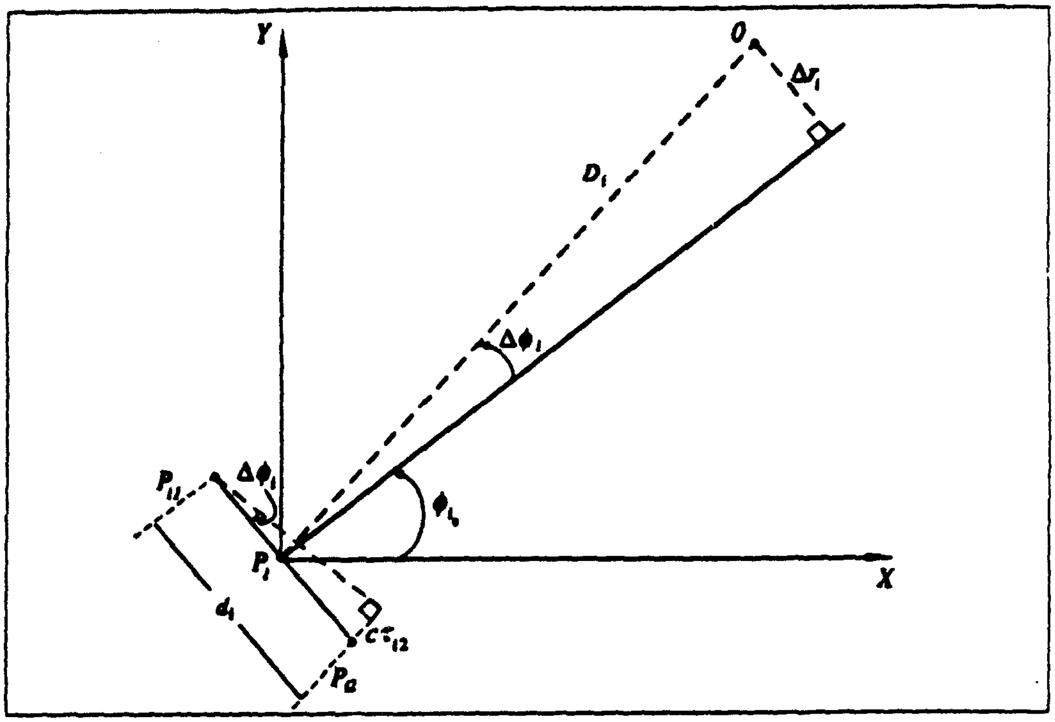
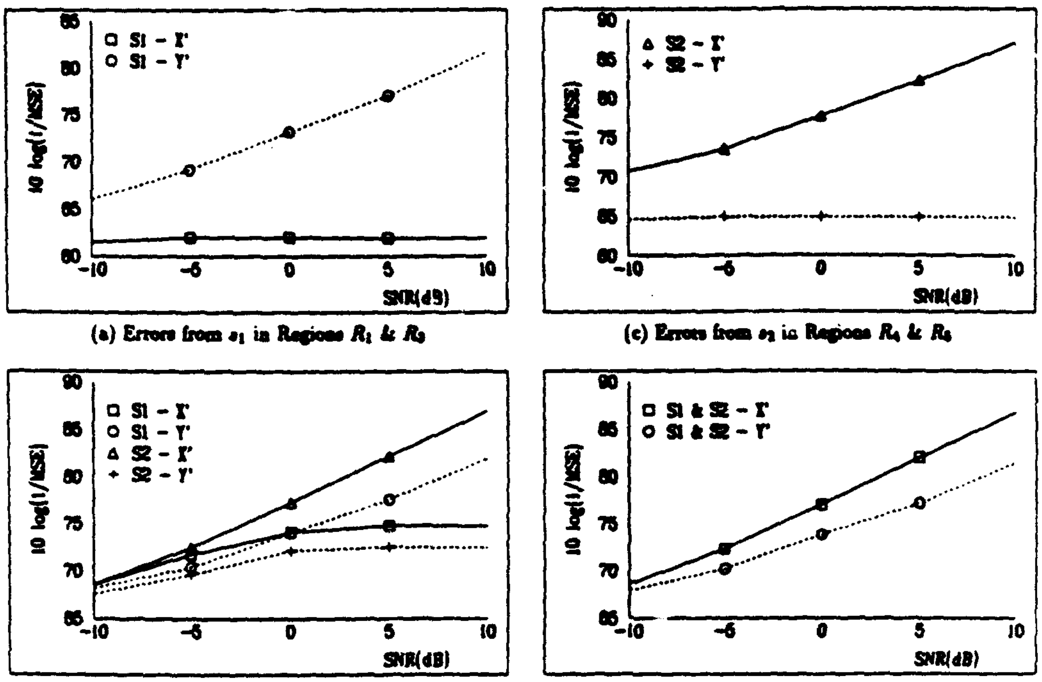


Figure 3 Single Sensor Intercept Geometry



(a) Errors from s_1 in Regions R_1 & R_3

(c) Errors from s_2 in Regions R_4 & R_5

(b) Errors from s_1 and s_2 in Region R_2

(d) Combined Errors from s_1 and s_2 in Region R_2

Figure 4 Multisensor Fusion In-Track and Cross-Track Estimation Errors

CENTRALIZED AND DISTRIBUTED HYPOTHESIS TESTING WITH STRUCTURED ADAPTIVE NETWORKS AND PERCEPTRON-TYPE NEURAL NETWORKS

Stelios C. A. Thomopoulos, Ioannis N. M. Papadakis, Haralambos Sahinoglou, and Nickens N. Okello

Decision and Control Systems Laboratory
Department of Electrical and Computer Engineering
The Pennsylvania State University
University Park, PA 16802
Tel.(814)865-3744:e.mail:sct@psuecl.bitnet

ABSTRACT

Two different types of adaptive networks are considered for solving the centralized and distributed hypothesis testing problem. The performance of the two different types of networks is compared under different performance indices and training rules. It is shown that training rules based on the Neyman-Pearson criterion outperform error based training rules. Simulations are provided for data that are linearly and nonlinearly separable.

I. INTRODUCTION

The optimum Bayesian and Neyman-Pearson solution to the distributed decision fusion problem bears striking similarities to the structure of a neural network (NN), [28,29]. Moreover, NNs can, in principle learn arbitrary input-output mappings, provided that they are sufficiently smooth. These two facts motivate the use of NNs for solving the centralized and distributed hypothesis testing problem. In selecting the proper NN layout, one could argue that a perceptron-type NN can learn any input-output mapping, thus it can be trained to solve the hypothesis testing problem. However, the ability of a perceptron-type NN to learn an arbitrary I/O mapping critically depends on the number of layers, the number of neurons per layer, and their interconnections which cannot, in general, be determined a priori.

In order to conduct a comprehensive study of the ability of adaptive networks to solve the centralized and distributed hypothesis testing (CHT and DHT) problem, two different types of adaptive networks are considered: structured adaptive networks (SANs) and perceptron-type neuron networks (PTNNs). By SAN we mean a network whose inputs are functionally related to the data through known functional transformations, and the outputs are parametrically dependent on the input. By PTNN we mean a multi-layered NN that consists of neurons in the classical sense, interconnected through synaptic weights.

The selected networks are trained using error based and Neyman-Pearson based indices of performance (IPs). The training rules are derived as gradient rules on the selected IPs. Simulations are conducted with linearly and nonlinearly separable Gaussian data.

II. Centralized Bayesian Hypothesis Testing (CBHT)

Assuming N statistically independent data sources, the optimal Bayesian or Neyman-Pearson (N-P) CBHT is the Likelihood Ratio Test (LRT)

$$\Lambda(r) = \Lambda(r_1, ..., r_N) = \prod_{i=1}^{N} \frac{dP(r_i|H_1)}{dP(r_i|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} T_f \qquad (II.1)$$

where r_i designates the data from the i-th sensor, H_i is the i-th hypothesis, $i = 0, 1$. The threshold T_f, for the Bayesian processor in determined by

$$T_f = \frac{P_0(C_{10} - C_{00})}{P_1(C_{01} - C_{11})} \qquad (II.2)$$

where P_0, $P_1 = 1 - P_0$ are the priors on the two hypotheses and C_{ij} is the cost of deciding in favor of hypothesis H_i when the true hypothesis is H_j, $i,j = 0,1$. For the N-P solution, the threshold T_f is determined by the false alarm requirement at the fusion according to

$$\int_{T_f}^{\infty} dP(\Lambda(r)|H_0) \leq \alpha_0 \tag{II.3}$$

where α_0 is the desired aggregate probability of false alarm (PFA) at the fusion. Notice that the Bayesian processor requires the knowledge of the priors (P_0, P_1) which may not be objectively available. The N-P processor circumvents this requirement by constraining the PFA and maximizing the probability of detection (PD). Also notice that both processors are parametric.

III. Distributed Binary Hypothesis Testing (DBHT)

Assuming that each sensor makes binary or multi-level independent decisions u_i, $i = 1, \ldots, N$, the optimal Bayesian or N-P DBHT solution under statistical independence consists of multilevel likelihood ratio quantizers (LRQs) [12,18] at each sensor and an LRT at the fusion. For binary LRQ at each sensor [4 to 19 and 22 to 31] with

$$u_i = \begin{cases} +1, & \text{if the i-th local decision favors hypothesis } H_1; \\ -1, & \text{if the i-th local decision favors hypothesis } H_0 \end{cases} \tag{III.1}$$

for the i-th sensor, the optimal Bayesian or N-P DBHT takes on the form

$$\sum_{i=1}^{N}(w_i u_i + t_i) \underset{H_0}{\overset{H_1}{\gtrless}} t_f \tag{III.2}$$

where

$$w_i = \frac{1}{2} log \frac{P_{D_i}(1 - P_{F_i})}{P_{F_i}(1 - P_{D_i})} \quad \text{and} \quad t_i = \frac{1}{2} log \frac{P_{D_i}(1 - P_{D_i})}{P_{F_i}(1 - P_{F_i})} \tag{III.3}$$

The threshold t_f for the Bayesian DBHT is determined by an expression similar to (II.2) that depends on the priors (P_0, P_1). For the N-P DBHT the threshold t_f is determined by the PFA requirement, equation (II.3). It is interesting to notice that (III.2) can be written as

$$\sum_{i=1}^{N} w_i u_i - t_0 \underset{H_0}{\overset{H_1}{\gtrless}} 0 \tag{III.4}$$

where

$$t_0 = -t_f - \sum_{i=1}^{N} t_i \tag{III.5}$$

The form of (III.4) is reminiscent of a NN, figures 1 and 2, [28,29].

IV. Centralised Hypothesis Testing and Distributed Decision Fusion with Structured Adaptive Networks (SANs)

A. Centralized Binary Hypothesis Testing with SANs

As discussed in Section II, the optimal decision test for a binary hypothesis problem is a likelihood ratio test (LRT) of the form

$$\Lambda(r) = \frac{p(r|H_1)}{p(r|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \eta \tag{IV.1}$$

where $p(.|H_i)$ is the conditional probability density function (pdf) of the data conditioned on H_i (i=0,1) and $\eta(\geq 0)$ is a threshold. For Gaussian problems, $ln[\Lambda(r)]$ has a simpler form and can be used in lieu of (IV.1) in the equivalent log-LRT

$$ln[\Lambda(r)] = ln\left[\frac{p(r|H_1)}{p(r|H_0)}\right] \underset{H_0}{\overset{H_1}{\gtrless}} \gamma := ln(\eta) \tag{IV.2}$$

For example, if the problem is of the form

$$r = \begin{cases} N(m_1, \sigma_1^2) & :H_1 \\ N(m_0, \sigma_0^2) & :H_0 \end{cases} \tag{IV.3}$$

where $N(m, \sigma^2)$ indicates a Gaussian pdf with mean m and variance σ^2, then the log-LRT test from (IV.2) gives

$$l(r) = \left[\frac{1}{\sigma_0^2} - \frac{1}{\sigma_1^2}\right] r^2 + 2\left[\frac{m_1}{\sigma_1^2} - \frac{m_0}{\sigma_0^2}\right] r \underset{H_0}{\overset{H_1}{\gtrless}} 2\gamma + \frac{m_1^2}{\sigma_1^2} - \frac{m_0^2}{\sigma_0^2} - 2ln\left[\frac{\sigma_0}{\sigma_1}\right] \tag{IV.4}$$

where $l(r)$ is the sufficient statistic for the problem (IV.3). The previous example serves as motivation for the structure of the network that is discussed in the following section.

1. Network Structure

The structure of the network is shown in Fig. 3. The functions ϕ_i are chosen to reflect any a priori knowledge about the problem. In the Gaussian problem for example, in view of (IV.4), it is natural to take

$$\phi_i(x) = x^i, \quad i = 0, 1, ..., k. \tag{IV.5}$$

with $k = 2$. In the general case $k \geq 2$. Note that in a general problem, the ϕ_i's can assume different functional forms. From figure 3, the output, y_j, of the network due to the data r_j is given by

$$y_j = g\left[\sum_{i=0}^{k} c_i \phi_i(r_j)\right] \tag{IV.6}$$

where $g(.)$ is a sigmoid function defined as

$$g(x) = \frac{1 - e^{-\lambda x}}{1 + e^{-\lambda x}} \tag{IV.7}$$

where $\lambda > 0$ adjusts the steepness of its slope. The network of figure 3 is capable of decision making, if one maps $y \geq 0$ to, say, H_1.

Given the above network structure, the hypothesis testing problem takes on the following form: given a set of ϕ_i's, $i = 0, 1, ..., k$, and a set of observations r along with the hypotheses under which they are generated, choose the coefficients $c_i, i = 0, 1, ..., k$, so that the resulting decision scheme is close to the optimal one in some suitably defined sense. It is therefore necessary to establish a criterion of optimality and an algorithm that updates the weights $c_i, i = 0, 1, ..., k$, in order to meet this criterion. The second task is the so called training of the network. In the sequel we discuss two different performance criteria and derive the update equations for the parameters of the network (synaptic weights) for each one of them.

The first criterion which appears more intuitive especially in view of the backpropagation method [20], is to minimize the sum of the squares of errors over all the training data. In this case, the index of performance (IP) can be defined by

$$J(t) = \sum_{j=1}^{N} \left[e_j(t)\right]^2 \tag{IV.8}$$

where N is the number of available training data (typically around 50-100 per hypothesis) and e_j is the error defined by

$$e_j(t) := y_j(t) - y_j^d, \quad j = 1, ..., N \tag{IV.9}$$

where y_j^d is equal +1 if r_j is generated under H_1 or -1 if it is generated under H_0. Note that the time index is introduced to denote updates of the weights c_i. Since (IV.8) does not impose any penalty on the relative magnitudes of the weights, a natural extension of (IV.8) is

$$J(t) = \sum_{j=1}^{N} \left[e_j(t)\right]^2 + \sum_{n=0}^{k} \rho_n c_n^2 \tag{IV.10}$$

where $\rho_n \geq 0$ are suitably chosen weighing coefficients. Under (IV.8) or (IV.10), the network will approximate a minimum probability of error classifier, i.e. will minimize the probability of error given by

$$P_E = Pr(H_1|H_0)P_0 + Pr(H_0|H_1)P_1 \tag{IV.11}$$

where P_0, P_1 are the prior probabilities of the respective hypotheses. In this case, the training will try to "fit" the model (IV.6) to the training data so that the sum of the square errors is minimized. Although this approach seems natural, it is not suitable for hypothesis testing problems for two reasons. First, the network that minimizes (IV.8) or (IV.10) for a given training set is not asymptotically optimal as the volume of the available training data goes to infinity simply because even if P_E can be made to be very close to zero for a given training set, (for example by taking $k \approx N$) the network may not result to P_E close to the probability of error of the LRT over the entire data ensemble. (Note that since similar data may be generated by either hypothesis, $P_E = 0$ is not always possible.) On the other hand, if k is kept moderate, fitting is very difficult especially when the data under both hypotheses are closely clustered as in the Gaussian case when the pdf's under the two hypotheses have the same mean and comparable variances. An additional problem with the training rule (IV.8) or (IV.10) is the lack of a general stopping criterion for the training. From the discussion above, (IV.8) and (IV.10) are not satisfactory criteria for our problem, although, they result in acceptable performance in linearly separable data cases as is shown in the simulations section.

The second criterion used for training is based on the Neyman-Pearson (N-P) approach which maximizes the probability of detection at a given (fixed) false alarm probability level. The key difference between the N-P and the the least squares error approach is that in the N-P training the hypotheses are separated and enter separately in the performance index. For this method, the performance index is given by

$$J(t) = \tilde{P}_M(t) + \frac{\rho}{2}[\tilde{P}_F(t) - P_{F_0}]^2 \quad (\rho \geq 0) \tag{IV.12}$$

where P_{F_0} is the preset level of false alarm probability and $\tilde{P}_M$, $\tilde{P}_F$ are defined by

$$\tilde{P}_M(t) := \frac{1}{2} \frac{\sum_{j=1}^{N}[1 + y_j^d][1 - y_j(t)]}{N + \sum_{j=1}^{N} y_j^d} \tag{IV.13}$$

$$\tilde{P}_F(t) := \frac{1}{2} \frac{\sum_{j=1}^{N}[1 - y_j^d][1 + y_j(t)]}{N - \sum_{j=1}^{N} y_j^d} \tag{IV.14}$$

and are approximate expressions for the miss probability P_M and the false alarm probability P_F of the network respectively. For a large sample size and large λ, the expression on the RHS of (IV.13) and (IV.14) approximate the $P_M(t)$ and $P_F(t)$ of the network. In view of (IV.12), the training in this case should compute the weights $c_i, i = 0, ..., k$, that minimize J for the given training set.

In the following, for each of the above optimality criteria, we derive the update equations for the (synaptic) weights.

2. Gradient Update Laws

The derivative of $g(x)$ is given by

$$g'(x) = \frac{2\lambda e^{-\lambda x}}{[1 + e^{-\lambda x}]^2} \qquad (IV.15)$$

The time derivative of $J(t)$ from (IV.8) is

$$\frac{dJ}{dt} = 2\sum_{j=1}^{N}\left\{e_j(t)\frac{de_j}{dt}\right\} = 2\sum_{j=1}^{N}\left\{e_j(t)\left[\sum_{n=0}^{k}\frac{\partial e_j}{\partial c_n}\frac{dc_n}{dt}\right]\right\} \qquad (IV.16)$$

from which it is clear that if

$$\frac{dc_n}{dt} = -a\sum_{j=1}^{N} e_j(t)\frac{\partial e_j}{\partial c_n} \quad (a > 0) \qquad (IV.17)$$

we have that

$$\frac{dJ}{dt} = -2a\sum_{n=0}^{k}\left[\sum_{j=1}^{N} e_j\frac{\partial e_j}{\partial c_n}\right]^2 \leq 0$$

which implies that J is decreasing for as long as the network does not reach an equilibrium point. A simple first order update expression for the weights follows directly from (IV.17) and from the fact

$$\frac{\partial e_j}{\partial c_n} = g'\left(\sum_{i=0}^{k} c_i\phi_i(r_j)\right)\phi_n(r_j) \qquad (IV.18)$$

and has the following form

$$c_n(t+1) = c_n(t) - (a\Delta t)\sum_{j=1}^{N} e_j(t)\left\{g'\left(\sum_{i=0}^{k} c_i\phi_i(r_j)\right)\right\}\phi_n(r_j) \qquad (IV.19)$$

where $n = 0, 1, ..., k$.

For (IV.10), in a similar manner, the recursion update laws are given by

$$c_n(t+1) = (1 + \rho_n\Delta t)c_n(t) - (a\Delta t)\left[\sum_{j=1}^{N} e^j g'\left(\sum_{i=0}^{k} c_i\phi(r_j)\right)\phi_n(r_j)\right] \qquad (IV.20)$$

which results in significant improvement on performance and rate of convergence as found from simulations.

For the IP given by (IV.12), the derivation of the update equations is as follows:

$$\frac{dJ}{dt} = \frac{d\tilde{P}_M}{dt} + \rho(\tilde{P}_F - P_{F_0})\frac{d\tilde{P}_F}{dt} \qquad (IV.21)$$

Using the chain rule, we obtain

$$\frac{d\tilde{P}_M}{dt} = \sum_{n=0}^{k}\frac{\partial\tilde{P}_M}{\partial c_n}\frac{dc_n}{dt}, \quad \frac{d\tilde{P}_F}{dt} = \sum_{n=0}^{k}\frac{\partial\tilde{P}_F}{\partial c_n}\frac{dc_n}{dt} \qquad (IV.22)$$

The partial derivatives in (IV.22) are given by the expressions

$$\frac{\partial \tilde{P}_M}{\partial c_n} = -\frac{1}{2}\frac{\sum_{j=1}^{N}(1+y_j^d)\frac{\partial y_j}{\partial c_n}}{N+\sum_{j=1}^{N} y_j^d} \tag{IV.23}$$

$$\frac{\partial \tilde{P}_F}{\partial c_n} = -\frac{1}{2}\frac{\sum_{j=1}^{N}(1-y_j^d)\frac{\partial y_j}{\partial c_n}}{N-\sum_{j=1}^{N} y_j^d} \tag{IV.24}$$

where as before

$$\frac{\partial y_j}{\partial c_n} = \left[g'\left(\sum_{i=0}^{k} c_i\phi_i(r_j)\right)\right]\phi_n(r_j) \tag{IV.25}$$

Hence the gradient update rule is given by

$$\frac{dc_n}{dt} = -a\left[\frac{\partial \tilde{P}_M}{\partial c_n} - \rho(\tilde{P}_F - P_{F_0})\frac{\partial \tilde{P}_F}{\partial c_n}\right] \tag{IV.26}$$

which results in the following iterative update expression for c_n

$$c_n(t+1) = c_n(t) - (a\Delta t)\left[\frac{\partial \tilde{P}_M}{\partial c_n} - \rho(\tilde{P}_F - P_{F_0})\frac{\partial \tilde{P}_F}{\partial c_n}\right] \tag{IV.27}$$

which in view of (IV.23), (IV.24) is a so-called batch training method since all training data are required for each update.

In the remainder of this section, we compare the performance of the above training methods for two hypothesis testing problems.

3. Simulation Results: The Centralized Case

The different hypothesis testing paradigms were selected in order to compare the performance of SANs in linearly and nonlinearly separable data ensembles under the MSE and N-P training rules. The performance was benchmarked with respect to the size of the training data ensemble, the number of power terms (ϕ_i's) in the functional representation of the data, and the training rule.

The two selected problems for centralized and distributed hypothesis testing were:

(i) a Linear Gaussian Problem (LGP)

$$r = \begin{cases} 1 + N(0,1) & :H_1 \\ N(0,1) & :H_0 \end{cases} \tag{LGP}$$

(ii) a Quadratic Gaussian Problem (QGP)

$$r = \begin{cases} N(0,5) & :H_1 \\ N(0,1) & :H_0 \end{cases} \tag{QGP}$$

where $N(m, \sigma^2)$ is the Gaussian distribution with mean m and variance σ^2. For each problem, the optimal LRT test follows directly from (IV.4).

In all cases, both the mean-squared-error (MSE) rule, eq. (IV.8), and the Neyman-Pearson (N-P) rule, eq.(IV.12), were used to train the SANs. The simulations were conducted as follows. The number of coefficients were fixed to either three (k=2) or six (k=5). Experiments with samples of one hundred (fifty per hypothesis) and two hundred (one hundred per hypothesis) data points were performed. The initial value of the c_i coefficients was zero in all simulations. For the MSE training, selective training was used to avoid convergence problems that arise during training from data that belong to different hypotheses but are "metrically" close. According to the selective rule, at each training, corrections were made only over those data points that were identified as belonging to the correct hypothesis at the beginning of the session.

An arbitrary stopping rule was also used to terminate the MSE training when the gradient was less than 10^{-5}.

N-P training was performed at different PFA's. The post-training Receiver Operating Characteristics (ROCs) were obtained by keeping all the c_i coefficients fixed at their training values and varying the threshold (c_0).

The ROCs were experimentally obtained by running ten thousand data points (five thousand per hypothesis) through the SAN but excluding the data points used for training. For each problem, we selected the coefficients that corresponded to the value of the PFA which generates the experimental ROC with the larger area when tested on the training data. For the LGP, the N-P training method outperforms the error training method. This is also the case for the QGP. The simulation results for both problems are summarized in Table 1 for the error training and Table 2 for the Neyman-Pearson method respectively.

Some conclusions drawn from the simulations follow.

1) The N-P training method outperforms the error based training method. This is clear from the QGP where the data under the two hypotheses are not well separated spatially as in LGP, in which the data are clustered around the two well separated means.

2) If the model is overparameterized, the performance of the NP-trained SAN is sensitive to the value of P_{F_0}. For example in the (QGP), the performance is good for $P_{F_c} = 0.7$ and poor for $P_{F_c} = 0.2$. As a result one should try several values of P_{F_c} and choose that one for which the ROC (obtained from testing on the training data after training) gives the ROC with the largest area. Furthermore, one could also start with a low value for k (say $k = 2$) and keep increasing its value, choosing finally the ROC with the largest area.

3) In general, N-P training results in a SAN that performs close to the optimum test. Since no a priori knowledge for the pdfs is necessary, this is a powerful approach especially in the case in which the volume of the available data is not sufficiently large for a reliable estimate of the pdfs under each hypothesis.

B. Distributed Decision Fusion with N-P Rule Trained SANs

1. Network Structure

The fusion system in Fig. 12, which consists of three identical sensors interconnected in parrallel was used to test the performance of N-P trained SANs in data and decision fusion problems. In the centralized data fusion test, each sensor in the configuration of Fig. 12 simply relays its observations to the fusion directly. The fusion is replaced by a SAN similar to the one shown in Fig. 3. Thus, the centralized data fusion SAN is identical to the one discussed in the previous section, except that three data are available at a time, instead of a single measurement as in the case of single sensor SAN.

In the distributed decision fusion (DDF), each sensor in the configuration of Fig. 12 is replaced by a SAN similar to the one Fig. 3. Due to the similarity of the sensors, it is assumed that a symmetric solution, i.e. identical synaptic weights and thresholds among all three sensors results in a solution that is close to the optimal one, if not "the optimal". Under the assumption (or constraint) of identical operating points, the structure of the optimal DDF, eq. (VI.2), simplifies to

$$\sum_{i=1}^{N} u_i \underset{H_0}{\overset{H_1}{\gtrless}} lnT_f \qquad (IV.28)$$

with the convention.

$$u_i = \begin{cases} 1 & \text{if the i-th local decision favors hypothesis } H_1 \\ 0 & \text{if the i-th local decision favors hypothesis } H_0 \end{cases} \qquad (IV.29)$$

Notice that the numerical values associated with each sensor decision are merely an expressional convenience and do not play any role in the outcome of the fusion process (see Section V as well).

Given the structure of the optimal DDF equation (IV.28) in the symmetric case, the only variables that determine the performance of the fusion for a target false alarm probability are the thresholds at the sensors (common among all sensors) and the fusion threshold. Thus, in the SAN implementation of the symmetric DDF only two parameters are adaptively adjusted: the common threshold for all sensors and the fusion threshold. This structure was used for training the SAN to perform the DDF for the fusion system of figure 12 using the N-P training rule. However, N-P training of the network by varying the two thresholds simultaneously resulted in very poor performance of the fusion. Thus, instead of training all the sensors simultaneously by minimizing the N-P performance index at the fusion, the ROC of each sensor was obtained separately using N-P training first. Then, the fusion rule was fixed a priori, and the network ROC was obtained by varying only the common threshold at the sensors after they were trained.

2. Simulation Results

In order to compare the performance of the centralized hypothesis testing with the DDF using the SAN, the same two binary hypothesis testing problems that were used for testing the performance of SANs in CBHT were also used for DBHT. The simulations for all problems were performed as follows: In all cases, the size of the training set is not larger than 200 data points. Post-training testing is performed on at least 2000 data points other, of course, than the training data points. The initial value of all c_i's is zero. Due to the training rules that implement a true gradient decent, convergence is monotonic in all cases.The values of the weights after training for each case are given in Table 2.

The DDF was done by pretraining each sensor with the test set individually using N-P training. To implement the ROC of each sensor, a SAN with two terms in the power expansion ($K = 2$) was used . For the LGP case 1, Table 2, the training set consists of 50 data points per hypothesis. The network was trained using 1000 iterations and the N-P training rule. For the QGP, 100 data points per hypothesis were used for training, case 3, Table 2. Since all the sensors are assummed to be identical and operating at the same operating false alarm and detection probability point, the synaptic weights (coefficients c_i) for the DDF for all three of them are identical, and identical to the weights used for hypothesis testing by each one individually, Table 2.

In all DDF cases, the sensors were assumed to be identical, all operating at the same PFA and P_D. The "OR", "AND", and the "ML" (majority logic) rules were used for decision fusion. The ROC of the different fusion rules for the DDF are compared among themselves and with the centralized fusion ROCs in Figs. 13, 14. The following conclusions can be drawn from these figures.

In the LPG, the majority rule seems to give the best ROC for DDF, which is close to the SAN performance on the centralized hypothesis testing. For the QGP, however, the OR rule seems to yield the best ROC, which again, is close to the centralized ROC. A general conclusion from the numerical results seems to be that for linear separable data, the majority fusion rule yields the best ROC. However, for quadratically separable data, the OR fusion rule yields the best ROC.

V. Distributed Decision Fusion with Perceptron-Type Neural Networks

Although the form of the optimal Bayesian/N-P DDF is known, for both binary and multi-level quantizations [9,12,14], the optimal thresholds are given, in general, in terms of coupled, nonlinear equations [8], [10], whose solution is not forthcoming even in simple cases. Suboptimal numerical solutions to the N-P DDF [10] may still be computationally intensive, if the fusion rule is unknown. The optimal solution to the Bayesian and Neyman-Pearson DDF problem, eq. (III.4) bears striking topological and functional similarities with the structure of a neural network (NN). This topological similarity suggests an alternative approach to solving the computationally N-P hard [5] DDF problem. By slightly modifying the values that designate the decision at the i-th sensor to

$$u_i = \begin{cases} +1 & \text{if the i-th local decision favors hypothesis } H_1 \\ 0 & \text{if the i-th local decision favors hypothesis } H_0 \end{cases} \tag{V.1}$$

for notational convenience, the optimal Bayesian and N-P DDF rule (III.4), takes on the form

$$\sum_i (w_i u_i + t_i) \underset{H_0}{\overset{H_1}{\gtrless}} T_f \tag{V.2}$$

where

$$w_i = log\left[\frac{P_{D_i}}{P_{F_i}}\right] - log\left[\frac{1-P_{D_i}}{1-P_{F_i}}\right] \tag{V.3}$$

and

$$t_i = log\left[\frac{1-P_{D_i}}{1-P_{F_i}}\right] \tag{V.4}$$

By combining the constant thresholds together with the unknown operational threshold T_f and defining

$$w_0 := -T_f + \sum_i t_i \tag{V.5}$$

the DDF rule (V.2) can be written in a form reminiscent of an NN architecture:

$$w_0 + \sum_i w_i u_i \underset{H_0}{\overset{H_1}{\gtrless}} 0 \tag{V.6}$$

A noticeable advantage of (V.6) over (V.2) is that the unknown threshold T_f has been absorbed in the synaptic weight w_0, which can be determined through training by assuming that it corresponds to the interconnection weight of an additional, constant input to the fusion neuron. Notice that the threshold in (V.6) is known, constant, and equal to zero. Thus, (V.6) can be implemented by using an NN and replacing the hard threshold decision rule by a smoother sigmoidal nonlinearity [20,21, Nils '90, TPS '90].

In figure 1 the optimal Bayesian (N-P) DDF structure is shown when the local LR is linear on the data. If the (local) sensors and fusion in figure 1 are identified with neurons and the thresholds are replaced by continuous sigmoid functions, there is a one-to-one topological correspondence between the D-S DDF architecture and the simple, two layer NN of figure 2. The topological similarities suggest that one can take advantage of the learning capabilities of an NN and train it to solve the Bayesian DDF even when the channel statistics are not known. The solution to Bayesian DDF can be achieved by using any one of the available training rules. For example, if a quadratic error is defined at the fusion by squaring the difference between the actual hypothesis and the output of the fusion, a gradient based algorithm, such as backpropagation [20], can be used to update the synaptic weights, i.e. the coefficients of the LRTs in the Bayesian DDF.

Training of the NN with a quadratic error criterion will result in a minimum error computer, if trained properly. A quadratic error training attempts to fit the data in two different hypotheses by minimizing a distance criterion. However, if the data in the training set are numerically close under the two hypotheses, overtraining of the NN in order to achieve perfect discrimination of the data in the training set will result in poor post-training performance. To avoid performance degradation from overtraining, selective training has been used with excellent results. The drawbacks associated with overtraining in the quadratic error criterion can be avoided by using an N-P based optimality criterion, such as the minimization of the miss probability at the fusion for fixed false alarm probability. Such a training criterion results in an NN that implements the optimal N-P DDF. If the optimal Bayesian DDF is highly nonlinear, an NN with inputs polynomial functions of the data (polynomial network) can be used to solve the optimal Bayesian DDF. This approach corresponds to approximating the LRT by a truncated Taylor's series expansion or a Voltera series similar to the approach used in SANs, figure 3, for determining the coefficients for each power in the T.S.E.

A. Training Rules

1. Backpropagation based on mean-squared error

Let the training output of the network be u_0^n at the n-th iteration, while the training hypothesis is u_i^n. The backpropagation method trains the NN by minimizing the

$$\text{error energy} = \sum_n (u_0^n - u_i^n)^2. \tag{V.7}$$

where the summation is over all training data during a training cycle. To implement a true gradient descent using the nomenclature of the generalized delta rule [20], define for each neuron k the function

$$\delta_k = o_k(1 - o_k) \sum_{\text{all } j \text{ that } k \text{ leads to}} \delta_k w_{kj} \tag{V.8}$$

where o_j is the output of neuron j and w_{kj} is the current weight between node k and node j. The output node is a special case where

$$\delta_n = 2(u_0^n - u_i^n)u_0^n(1 - u_0^n) \tag{V.9}$$

The update of the weights during training is done using the difference equation

$$dw_{ij}^n = \eta \delta_j o_i + \alpha dw_{ij}^{n-1}, \tag{V.10}$$

where η and α are predefined constants that determine the rate of convergence. The second term in the weight update equation is known as the momentum term.

The NN that was used for DDF consisted of three identical sensors and a fusion. Each sensor was represented by an identical NN, each having one input neuron, one hidden layer with three neurons, and a single-neuron output layer. The fusion NN consisted of three input-layer neurons, three hidden-layer neurons and a single-neuron output layer. The NN was first trained on the LPG and QGP of the previous section.

Backpropagation was used to train the three layer neural network to perform DDF. The test for convergence was based on the criterion

$$\sqrt{\sum_{n=1}^{N} \left\{ \Big[\sum_{ij \text{ all weights}} (dw_{ij}^n)^2 \Big] \Big/ \Big[\sum_{ij \text{ all weights}} (w_{ij}^n)^2 \Big] \right\}} < 10^{-2} \tag{V.11}$$

Training was terminated when the criterion (V.11) was satisfied.

2. Training based on Neyman-Pearson

N-P training is conceptually identical to the backpropagation algorithm, except that training is done around a desired false alarm rate at the fusion. In order to achieve training around a desired false alarm rate α at the fusion, two possible performance criteria can be used to measure the output error:

$$E = P_M + \lambda(P_F - \alpha)^2 \tag{V.12}$$

or

$$E = P_M^2 + \lambda(P_F - \alpha)^2 \tag{V.13}$$

where P_M, P_F are the miss and false alarm probabilities at the fusion.

The modifications required to the standard backpropagation to implement the N-P fusion rule relate only to the energy function derivative with respect to the output. To get this, first we express the probabilities in terms of the output as

$$P_M = \frac{\sum_{n=1}^{N}(1 - u_0^n)u_i^n}{\sum_{n=1}^{N} u_i^n} \tag{V.14}$$

$$P_F = \frac{\sum_{n=1}^{N}(1 - u_i^n)u_0^n}{\sum_{n=1}^{N}(1 - u_i^n)} \tag{V.15}$$

which give two possible derivative forms

$$\frac{dE}{du_o^m} = -\frac{u_i^m}{\sum_{n=1}^{N} u_i^n} - 2\lambda(P_F - \alpha)\frac{(1 - u_i^m)}{\sum_{n=1}^{N}(1 - u_i^n))} \tag{V.16}$$

$$\frac{dE}{du_0^m} = -2P_m\frac{u_i^m}{\sum_{n=1}^{N} u_i^n} - 2\lambda(P_F - \alpha)\frac{(1 - u_i^m)}{\sum_{n=1}^{N}(1 - u_i^n)} \tag{V.17}$$

for (V.12) and (V.13) respectively. If we set

$$\delta_o = \sum_{m=1}^{N} \frac{dE}{du_o^m} u_o^m(1 - u_o^m) \tag{V.18}$$

where "o" designates the output neuron, then the backpropagation rule proceeds as described above. The update rule (V.10) with δ_o defined by (V.18) implements a true gradient descent training by batch-processing the training set, whereas the backpropagation with δ_o defined by (V.9) implements a "pseudo"-gradient descent. A pseudo-gradient back propagation with the N-P energy functions (V.12) or (V.13) did not manage to produce a suitably trained NN. However, the true gradient N-P training rule (V.18) was successfully used in training the NN to solve the DDF problems.

3. Training based on Kalman Filter

The problem of training a NN can be viewed as a Kalman Filtering problem [23]. If the ideal (unknown) weights and thresholds of the NN are identified with the state $x(n)$ of a Kalman Filter, then these weights should be time-invariant, thus satisfying the plant equation.

$$x(n+1) = x(n) \tag{V.19}$$

The unknown state $x(n)$ in the NN is observed via the nonlinear output equation

$$d(n) = h(x(n)) + v(n) \tag{V.20}$$

where the error made from not knowing the weights and thresholds precisely is modeled as zero mean, random error $v(n)$ with covariance matrix $E[v(n)v(n)^T] = R(n)$, a positive definite matrix. The nonlinear function h(.) takes into account all the threshold nonlinearities at each neuron at every layer. From the nonlinear Kalman Filter theory, the state $x(n)$ can be estimated using the Extended Kalman Filter (EKF) with equations

$$x(n+1) = x(n) + K(n)[d(n) - h(x(n))] \tag{V.21}$$

$$K(n) = P(n)H(n)[R(n) + H^T(n)P(n)H(n)]^{-1} \tag{V.22}$$

$$P(n+1) = P(n) - K(n)H^T(n)P(n) \tag{V.23}$$

where $H(n)_{ij}$ is the derivative of the output i with respect to weight j, computed as in the backpropagation. Also $d(n)$ is the desired vector output neurons. For more datails on the use of the EKF for training the NN to perform the DDF see [22]

B. Simulation Results

The input data for each NN sensor were generated from the LGP and QGP distributions that were used to benchmark the SANs. The results are shown in figures 15 through 18. For the LGP one hundred training points were sufficient to obtain a ROC close to the optimal DDF. However, for the QGP, one thousand sample points were required to obtain acceptable ROC. If the solutions of the error based backpropagation are compared with the N-P based backpropagation, it is seen that the later results in superior performance. Yet if the results from the perceptron-type NN are compared with the N-P trained SAN, figures 13 and 14, the later results in superior performance with considerably fewer data samples, in particular for the QGP. (200 points for SAN vs 1000 points for PTNN). However, it should be stressed that no separate pretraining of each sensor NN was required with BPTNN, as was required for SANs in order to perform DDF.

Overall, SANs have the advantage that their performance can be understood and interpreted analytically since they are by construction parametric approximation to the LR optimal fusion rules. For the PTNNs, such an interpretation is not forthcoming, limiting the extrapolation of conclusions based on limited training data sets to general classes of problems.

VI. SUMMARY

Natural structural similarities between the Bayesian DDF solution and adaptive networks are exploited. It is shown that structured adaptive networks (SANs) and perceptron-type neuron networks (PTNNs) can learn to solve centralised and distributed hypothesis testing problems efficiently, even in the absence of explicit statistical information about the data, provided that the proper training rule and procedure are followed. Two training rules are invastigated: a mean squared error (MSE) based rule, and a rule based on the Neyman-Pearson (N-P) test. Under both training rules, the post-training performance of the network is very comparable to the optimal likelihood ratio test (LRT). However the N-P rule trained networks outperforms the MSE rule trained network, even when selective training is used with the later. The behavior of the networks under the two training rules is studied extensively in hypothesis testing problems with linearly and nonlinearly separable data. Similarities and differences in the behavior and performance of the networks are discussed.

REFERENCES

[1] Dempster, A.P., "A Generalization of Bayesian Inference," Journal of the Royal Statistical Society, Vol.30, 1968, pp.205-247.

[2] Van Trees, H. L., Detection, Estimation, and Modulation Theory, Vol.1, John Wiley & Sons. New York,1968.

[3] Shafer,G.A., A Mathematical Theory of Evidennce, Princeton University Press, 1976.

[4] Tenney, R.R. and Sandell, N.R., Jr., "Detection with Distributed Sensors," IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-17, July 1981, pp.501-510.

[5] Tsitsiklis, J.,and Athans, M., "On the Complexity of Distributed Decision Problem," IEEE Trans. on Automatic Control, AC-30, Vol.5, May 1985, pp.440-446.

[6] Sadjadi, F.A., "Hypothesis Testing in a Distributed Environment," IEEE Trans. on Aerospace and Electronic Systems, Vol.AES-22, March 1986, pp.134-137.

[7] Chair, Z. and Varshney,P.K., "Optimal Data Fusion in Multiple Sensor Detection System," IEEE Trans. on Aerospace and Electronic Systems, Vol.AES-22, No.1, January 1986, pp.98-101.

[8] Srinivasan. R.. "Distributed Radar Detection Theory," IEE Proceedings, Vol.133. Pt.F. No.1. February 1986, pp. 55-60.

[9] Thomopoulos, S.C.A., Viswanathan. R. and Bougoulias. D.K.. "Optimal Decision Fusion in Multiple Sensor Systems," IEEE Trans. on Aerospace and Electronic Systems, Vol.AES-23, No.5, Sept.1987.

[10] Thomopoulos. S.C.A.. Bougoulias. D.K., and Zhang,L., "Optimal and Suboptimal Distributed Decision Fusion." SPIE Proceedings, Vol.931, Sensor Fusion(1988), pp.26-30.

[11] Viswanathan.R., Thomopoulos, S.C.A., and Tumuluri, R., "Optimal Serial Distributed Decision Fusion," IEEE Trans. on Aerospace and Electronic Systems. Vol.24, No.4, pp.366-376, July 1988.

[12] Viswanathan. R., Ansari, A., and Thomopoulos, S.C.A., Optimal Partitioning of Observations in Distributed Detection, "Abstracts of Intern. Symposium on Information Theory, Kobe, Japan, June 19-24. 1988, pp.195. Also. Tech. Report. TR-SIU-DEE-3-87, Info. Proc. and Intell. Syst. ab., Southern Illinois Univ., 1987.

[13] Thomopoulos, S.C.A., and Okello, N.N., "Distributed Detection with Mismatched Sensors," SPIE'88 Symposium on Advances in Intelligent Robotic Systems, Boston, Nov.1988.

[14] Thomopoulos. S.C.A., Viswanathan.R. and Bougoulias, D.K., "Optimal and Suboptimal Distributed Decision Fusion," IEEE Trans. on Aerospace and Electronic Systems. Vol.25, No.5, Sept.1989.

[15] Thomopoulos. S.C.A., "Sensor Integration and Data Fusion, "Special Issue on Sensor Integration Data Fusion," International Journal of Robotics. Vol.7, No.3. 1990, pp.337-372.

[16] Thomopoulos, S.C.A., "Generalised Evidence Processing Theory," SPIE'90, Symposium on Advances in Intelligent Robotic Systems. Boston. Nov.5-9, 1990.

[17] Galuza. M., Performance Evaluation of Generalized Evidence Theory in Distributed Detection Problems, M.S. Thesis, Dept. of Electr. & Comp. Engr., Penn State Univ., May 1990.

[18] Tsitsiklis. J.N., "External Properties of Likelihood Ratio Quantizers," 1990 preprint.

[19] Longo, M.. Lookabaugh, T.D., and Gray, R.M., "Quantization for Decentralization Hypothesis Testing under Communication Constraints," IEEE Trans. on Info. Th., IT-36, No.2, March 1990, pp.241-255.

[20] McClleland. J.L., and Rumelhart, D.E., Parallel and Distributed Processing, MIT Press. Cambridge. MA. 1987.

[21] Nilsson, N.J., The Mathematical Foundation of Learning Machines, Morgan Kaufman Publishers. San Mateo, Ca, 1990.

[22] Thomopoulos. S.C.A., Papadakis, I.N.M., and Sahinoglou, H., "Centralized and Distributed Decision with Adaptive Networks," Tech. Report, PSU-DCSL-ECE-10-91, Oct. 1991.

[23] S. Singhal.and L. Wu, "Training Multilayer Perceptron with the Extended Kalman Algorithm," in Advances in Neural Network Information Processing Systems 1, Ed. D.S. Touresky, Morgam Kaufmann Publ.. Palo Alto, Ca. 1989.

[24] D.W. Ruck, S.K. Rogers, and P.S. Maybeck, Back Propagation: A Degenerate Kalman Filter?,"preprint.

[25] Papoulis, A., "Probability, Random Variables, and Stochastic Processes," McGraw Hill, New York. 1984. 2nd Ed.

[26] Tsitsiklis, J.N., "Decentralized Detection by a Large Number of Sensors," Math. Control Signals Systems. (1988) 1: pp.167-182.

[27] Thomopoulos, S.C.A., "Sensor Integration and Data Fusion," SPIE Vol.1198, Sensor Fusion II: Human and Machine Strategies (1989), pp.178-191.

[28] Thomopoulos, S.C.A.. "Theories in Distributed Decision Fusion: Comparison and Generalization." SPIE 1990, Symposium on Advances in Intelligent Robotic Systems. Boston, Nov.5-9, 1990.

[29] Thomopoulos, S.C.A., "Theories in Distributed Decision Fusion: Comparision and Generalization," Tech. Report, PSU-DCSL-ECE-8-90, Aug. 1990.

[30] Thomopoulos, S.C.A., and D.K. Bougoulias,"DIGNET: A Self Organizing Neural Network for Automatic Pattern Recognition and Classification," SPIE Proceedings on Sensor Fusion, Orlando, FL, April 1991.

31 Bougoulias, D.K., *Distributed Decision Making with Bayesian and Neural Networks Approaches*, Ph.D. Dissertation, Southern Illinois University, Dept. of Electr. Engr., Carbondale, IL, June 1991.

32 Bougoulias, D.K., and Thomopoulos, "Centralised and Distributed Decision Making using DIGNET," SPIE Proceedings on Sensor Fusion, Boston, MA, Nov.1991.

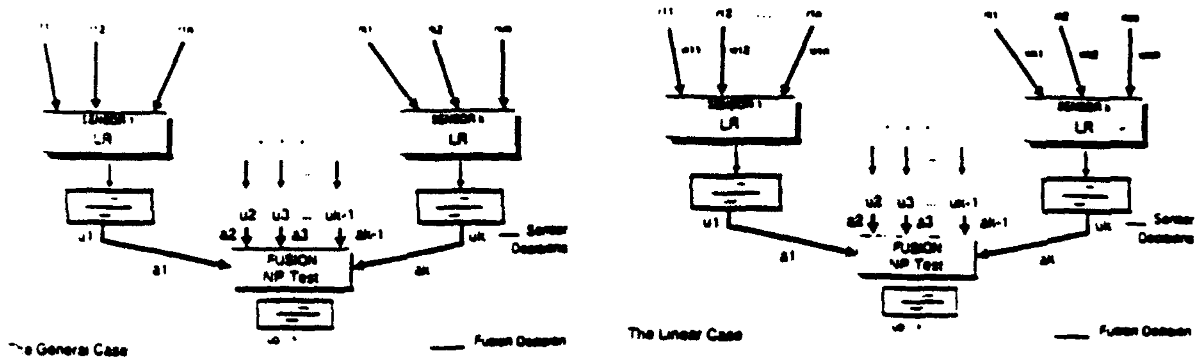


Figure 1. [illegible]

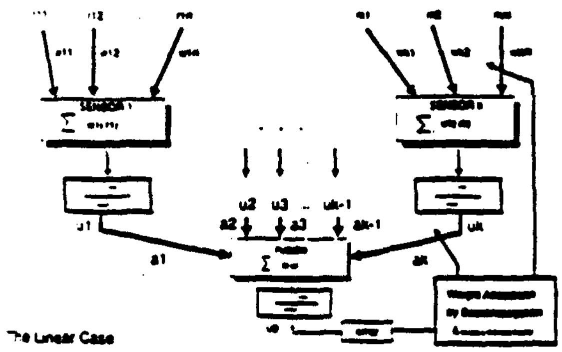


Figure 2. [illegible]

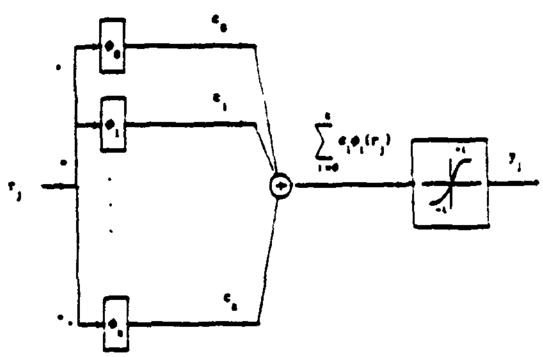


Figure 3. [illegible] (SAN) [illegible]

Case #	Problem	c_1	c_2	c_3	c_4	c_5	# of training data
1	LGP	.898	-.278	N/A	N/A	N/A	100
2	QGP	-.349	.399	N/A	N/A	N/A	100
3	QGP	-.168	.417	N/A	N/A	N/A	200
4	QGP	-.237	.506	.245	.194	-.847	100

Table 1. Error Training.

Case #	Problem	c_1	c_2	c_3	c_4	c_5	# of training data	P_{F_2}
1	LGP	.923	.370	N/A	N/A	N/A	100	5
2	QGP	-.260	.796	N/A	N/A	N/A	100	.5
3	QGP	-.158	.803	N/A	N/A	N/A	200	5
4	QGP	-.793	.280	-.257	.408	.116	100	[illegible]

Table 2. Neyman-Pearson Training.

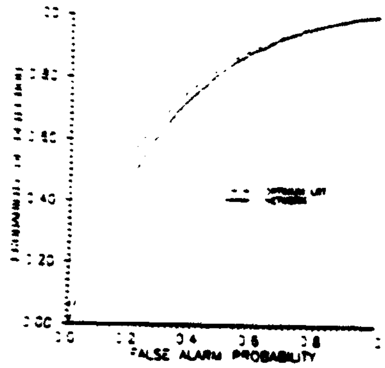


Figure 4. ROC for the linear gaussian problem (LGP) using SAN with error training with 100 training data and n=2, i.e case 1 of Table 1. The ROC for the LRT is also shown for comparison.

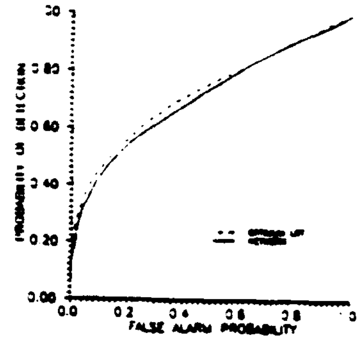


Figure 5. ROC for the quadratic gaussian problem (QGP) using SAN with error training with 100 training data and n=2, i.e case 2 of Table 1. The ROC for the LRT is also shown for comparison.

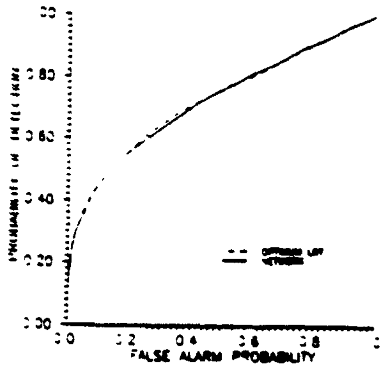


Figure 6. ROC for the quadratic gaussian problem (QGP) using SAN with error training with 200 training data and n=2, i.e case 3 of Table 1. The ROC for the LRT is also shown for comparison.

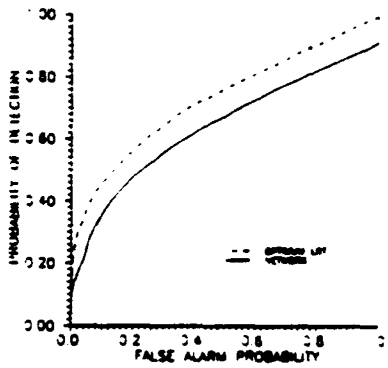


Figure 7. ROC for the quadratic gaussian problem (QGP) using SAN with error training with 100 training data and n=3, i.e case 4 of Table 1. The ROC for the LRT is also shown for comparison.

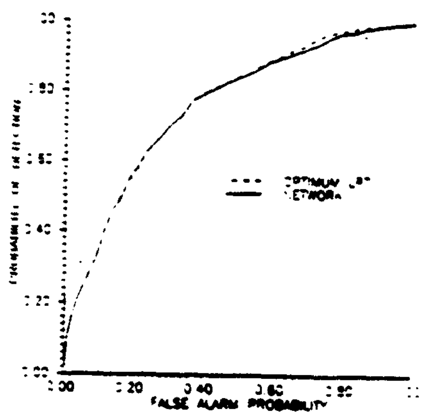


Figure 8. ROC for the linear gaussian problem (LGP) using SAN with Neyman-Pearson training with 100 training data, n=2 and preset false alarm (PFA) probability PFA=.5, i.e case 1 of Table 2. The ROC for the LRT is also shown for comparison.

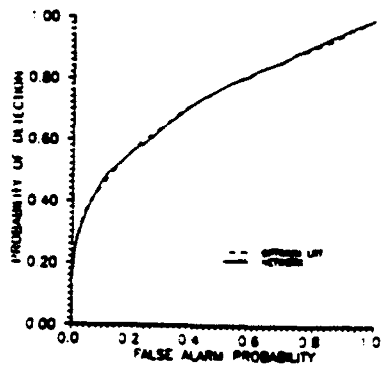


Figure 9. ROC for the quadratic gaussian problem (QGP) using SAN with Neyman-Pearson training with 100 training data, n=2 and preset false alarm (PFA) probability PFA=.5, i.e case 2 of Table 2. The ROC for the LRT is also shown for comparison.

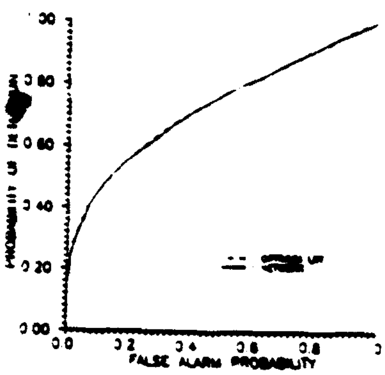


Figure 10. ROC for the quadratic gaussian problem (QGP) using SAN with Neyman-Pearson training with 100 training data, (n=2) and given false alarm (PFA) probability PFA=.5, i.e. case 3 of Table 2. The ROC for the LRT is also shown for comparison.

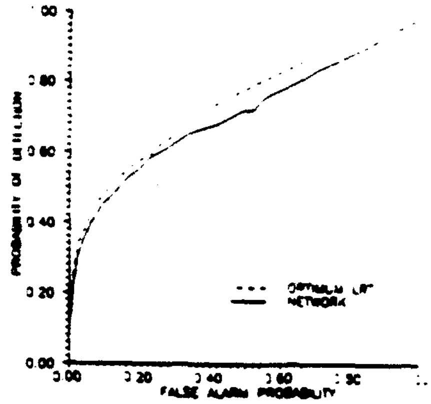


Figure 11. ROC for the quadratic gaussian problem (QGP) using SAN with Neyman-Pearson training with 100 training data, (n=3) and given false alarm (PFA) probability PFA=..., i.e. case 4 of Table 2. The ROC for the LRT is also shown for comparison.

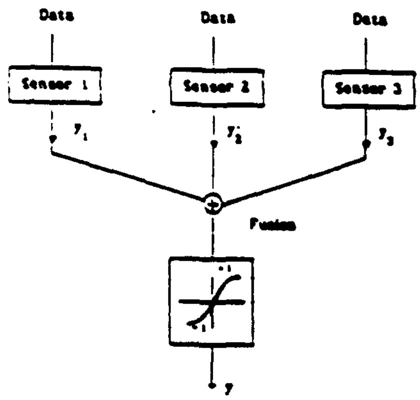


Figure 12. Structured Adaptive Network (SAN) for distributed hypothesis testing.

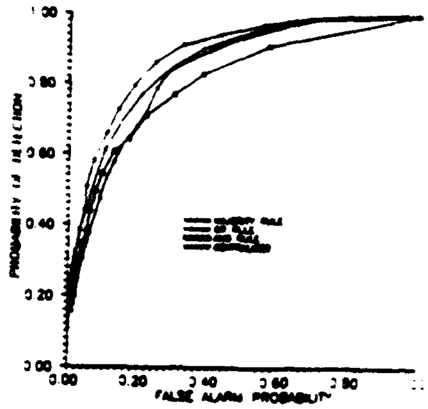


Figure 13. ROC for the linear gaussian problem (LGP) using SAN with 3 sensors, each having a ROC as shown in Figure 4, i.e. case 1 of Table 2, and fixed decision rules at the fusion center. The ROC of the centralized configuration is also shown for comparison.

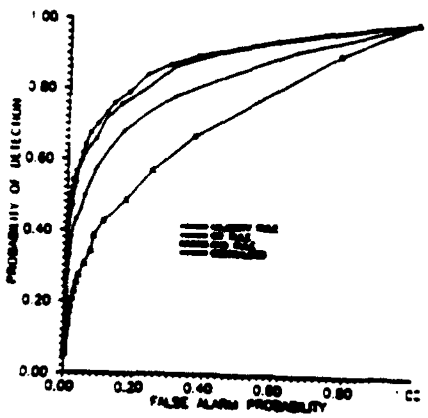


Figure 14. ROC for the quadratic gaussian problem (QGP) using SAN with 3 sensors, each having a ROC as shown in Figure 10, i.e. case 3 of Table 2, and fixed decision rules at the fusion center. The ROC of the centralized configuration is also shown for comparison.

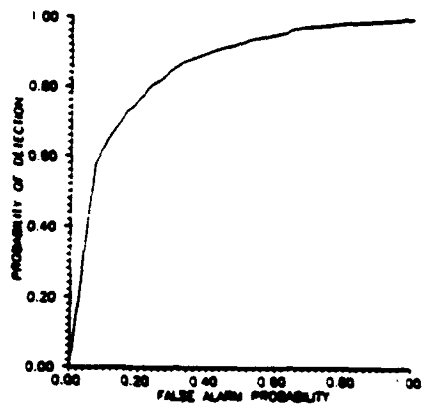


Figure 15. ROC for the linear gaussian problem (LGP) using Perceptron-Type Neural Network and MSE Backpropagation training with 3 identical sensors and 100 training data.

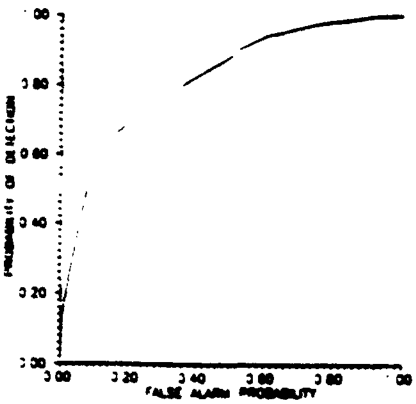


Figure 16. ROC for the linear gaussian problem (LGP) using Perceptron-Type Neural Network and N-P based gradient training with 3 identical sensors and 100 training data.

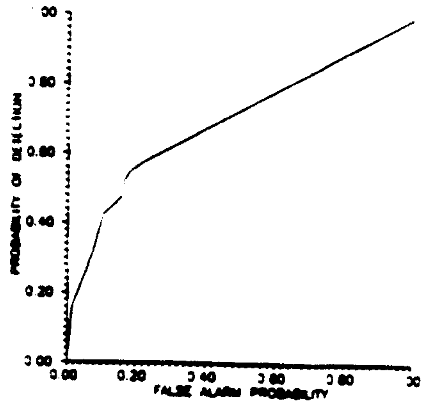


Figure 17. ROC for the quadratic gaussian problem (QGP) using Perceptron-Type Neural Network and MSE Backpropagation training with 3 identical sensors and 100 training data.

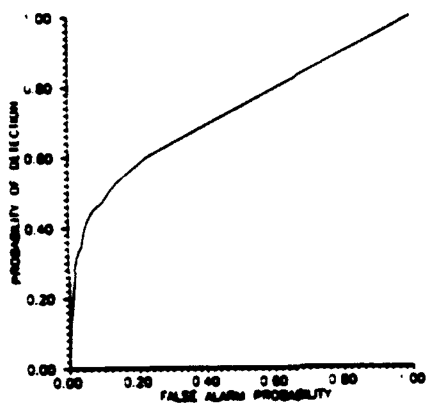


Figure 18. ROC for the quadratic gaussian problem (QGP) using Perceptron-Type Neural Network and N-P based gradient training with 3 identical sensors and 1000 training data.